ПРОГРАММИРОВАНИЕ НА SCRATCH



ПОДРОБНОЕ ПОШАГОВОЕ РУКОВОДСТВО ДЛЯ САМОСТОЯТЕЛЬНОГО ИЗУЧЕНИЯ РЕБЕНКОМ

ДЛЯ СРЕДНЕГО ШКОЛЬНОГО ВОЗРАСТА

Д.В. ГОЛИКОВ, А.Д. ГОЛИКОВ


SHOUT
SHARE
IT

- О ЧЕМ ЭТА КНИГА.
- КАК РАБОТАТЬ С КНИГОЙ.
- БЛАГОДАРНОСТИ.
- ССЫЛКИ.
- ВВЕДЕНИЕ.
- ГЛАВА 1. КАК УСТРОЕН SCRATCH. ПЕРВАЯ ПРОГРАММА.
- ГЛАВА 2. УСЛОЖНЯЕМ ПЕРВУЮ ПРОГРАММУ.
- ГЛАВА 3. ЦИКЛИЧЕСКОЕ ВЫПОЛНЕНИЕ ПРОГРАММЫ.
- ГЛАВА 4. ПЕРВЫЙ ПРОСТЕНЬКИЙ МУЛЬТФИЛЬМ.
- ГЛАВА 5. ФУТБОЛ.
- ГЛАВА 6. ЗНАКОМИМСЯ С КООРДИНАТОЙ X.
- ГЛАВА 7. ЗНАКОМИМСЯ С КООРДИНАТОЙ Y.
- ГЛАВА 8. МУЛЬТИК «ЛЕТУЧИЙ КОТ И ЛЕТУЧАЯ МЫШЬ».
- ГЛАВА 9. ИГРА ЛАБИРИНТ.
- ГЛАВА 10. МУЛЬТИК С ПРИВИДЕНИЯМИ.
- ГЛАВА 11. ИГРА «КОТЁНОК НА МИННОМ ПОЛЕ».
- ГЛАВА 12. ИГРА ПРО ВОЛШЕБНИКА.
- ГЛАВА 13. КОТ МАТЕМАТИК.
- ГЛАВА 14. ИГРА «КОТ С РЕАКТИВНЫМ РАНЦЕМ».
- ГЛАВА 15. ИГРА ПЛАТФОРМЕР.
- ГЛАВА 16. ИГРА «ЛОВИ ВКУСНЯШКИ».
- ГЛАВА 17. AUT VINCERE, AUT MORI. ПОБЕДА ИЛИ СМЕРТЬ!
- ПРИЛОЖЕНИЕ 1. ОСНОВНЫЕ ПОНЯТИЯ SCRATCH 2.
- ПРИЛОЖЕНИЕ 2. СПРАВОЧНИК ПО БЛОКАМ SCRATCH 2.0.
- ПРИЛОЖЕНИЕ 3. ОТРИЦАТЕЛЬНЫЕ ЧИСЛА.
- ПРИЛОЖЕНИЕ 4. ГРАДУСЫ.
- ПРИЛОЖЕНИЕ 5. КООРДИНАТЫ X И Y.
- ПРИЛОЖЕНИЕ 6. ДЕСЯТИЧНЫЕ ДРОБИ.
- ПРИЛОЖЕНИЕ 7. ПРОЦЕНТЫ.
- ПРИЛОЖЕНИЕ 8. ФУНКЦИИ.
- КРАТКОЕ СОДЕРЖАНИЕ СЛЕДУЮЩЕЙ КНИГИ.

Программирование на Scratch 2.

Делаем игры и мультики.

Подробное пошаговое руководство для самостоятельного изучения ребёнком.

Голиков Денис и Голиков Артём

Copyright Голиков Денис и Голиков Артём 2014

Al rights reserved.

Лицензионные Соглашения

Благодарю вас за загрузку этой бесплатной электронной книги. Хотя это - бесплатная книга, но она остается охраняемой

авторским правом собственностью автора, и не может копироваться и распространяться в коммерческих целях. Если Вам эта книга понравилась, пожалуйста, посоветуйте друзьям загрузить свою собственную копию. Благодарю вас за вашу поддержку.

О ЧЕМ ЭТА КНИГА.

Целью книги является пропедевтика программирования, математики, естественных наук среди школьников младших классов. Книга написана на основе опыта обучения программированию на Scratch в кружке юных программистов.

Материал рассчитан на самостоятельное изучение школьниками 3–5 классов, имеющими базовые навыки управления компьютером. Дети должны уметь пользоваться «мышью», запускать ярлыки и т.п., а также считать до 480, умножать и делить. Более сложные математические понятия (отрицательные числа, десятичные дроби, оси координат, градусы) будут объяснены на страницах книги.

Scratch — это новая бесплатная среда программирования, которая позволяет младшим школьникам создавать собственные истории, мультфильмы, игры, презентации и другие произведения. Scratch специально разрабатывался как новая учебная среда для обучения школьников программированию. Программы создаются из разноцветных блоков точно так же, как собираются из разноцветных кирпичиков конструкторы Лего.

Scratch разработан в 2006г. под руководством Митчела Резника группой Lifelong Kindergarten в лаборатории Media Lab Массачусетского технологического института.

КАК РАБОТАТЬ С КНИГОЙ.

Основной принцип обучения ребёнка программированию по этой книге состоит в самостоятельном создании школьником сложной игры, состоящей из многих небольших конструкций — скриптов. Принцип работы каждого скрипта изложен очень подробно. После создания игры, ученик может некоторое время поиграть в игру, и, тут же внести изменения в игровой процесс, упростить или усложнить игру.

После каждой главы даны небольшие вопросы, которые помогут ребёнку проверить, всё ли он понял в работе игры, а также несколько заданий по её доработке. Как Вы уже можете догадаться, не смотря на то, что каждая игра в книге является законченной программой, её совершенствование может продолжаться бесконечно. Это поможет начинающему программисту научиться работать над большими и сложными приложениями, поможет развить память и аналитический склад ума.

Не стоит забывать, что создание игр потребует от вашего школьника умения рисовать на компьютере. В ранних версиях Scratch существовал только простейший редактор изображений, похожий на стандартный редактор Paint, однако, в версии Scratch 2.0, о которой пойдёт речь в этом учебнике, к простейшему растровому редактору, к огромной радости детишек, добавлен векторный редактор. Это многое изменило, игры стали выглядеть опрятней, проще стало изменять персонажей, в процесс обучения добавилось больше веселья. Изучению и приёмам работы в векторном редакторе в книге уделено очень много внимания, не буду скрывать, некоторым ребятам создание векторных персонажей даётся нелегко, зато, после освоения приёмов рисования, позволяет хвастаться перед друзьями, профессионально сделанными играми. Стоит заметить, что все игровые персонажи легко модифицируются, и могут быть изменены учениками без уведомления автора. На занятиях в «Клубе юных программистов», только самые ответственные дети стремятся сделать персонажей точно, как в учебнике, большая часть пытается нарисовать всё по-своему. Если программирование поведения игровых объектов некоторым ребятам не позволяет с лёгкостью включать фантазию и сразу же модифицировать программу по своему усмотрению, программы всё же довольно сложные, то в процессе рисования фона и персонажей фантазируют почти все. Чего только не делали они с бедным котиком!

БЛАГОДАРНОСТИ.

Спасибо создателям Scratch за волшебную возможность заниматься творчеством в новом формате и обучать детей новейшим технологиям.

Огромная благодарность Марии Конопатовой за обложку и иконки Котёнка. Все её работы можно найти на сайте http://komixsisters.ru.

Спасибо ребятам, которые помогали тестировать главы книги.

Спасибо тем, кто не оставался равнодушным, и направлял критические замечания.

ССЫЛКИ.

Сайт авторов, посвящённый обучению детей программированию на Scratch. http://scratch4russia.com.

Все примеры из книги опубликованы на сайте http://scratch.mit.edu/users/scratch_book/

Вопросы авторам можно задавать по почте scratch.book@ya.ru.

Страничка в Фейсбук: http://www.facebook.com/ScratchBook4u

Страничка Вконтакте: http://vk.com/scratch.book

ВВЕДЕНИЕ.

Существует два способа работы в Scratch. Можно работать в браузере, а можно установить оффлайновый редактор и делать программы не подключаясь к интернет. Если Вы планируете работать используя интернет, то прочитайте первую часть введения. Если Вы планируете работать без подключения к интернет, то можете перейти сразу ко второй части введения.

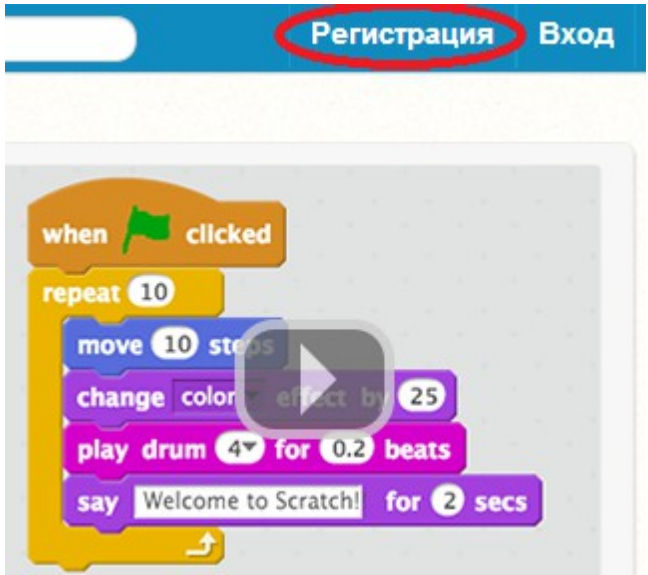


Обратите внимание, все примеры проектов в этой книге сделаны в оффлайновом редакторе.

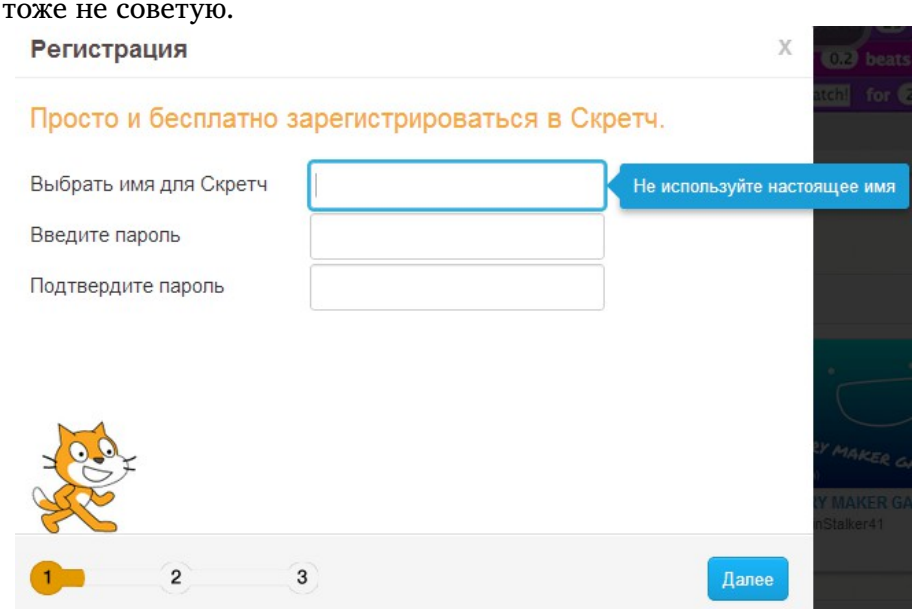
РАБОТА НА ПОРТАЛЕ SCRATCH.MIT.EDU.

Для получения возможности создавать свои проекты на Scratch, Вам необходимо зарегистрироваться на сайте scratch.mit.edu. Для регистрации Вам нужно иметь свой адрес электронной почты, попросите родителей Вам его завести. Когда у Вас появится свой адрес электронной почты, то первым делом запишите его в блокнот. Пароль тоже лучше записать, чтобы случайно не забыть.

Теперь перейдите на сайт scratch.mit.edu и нажмите на кнопку «Регистрация».



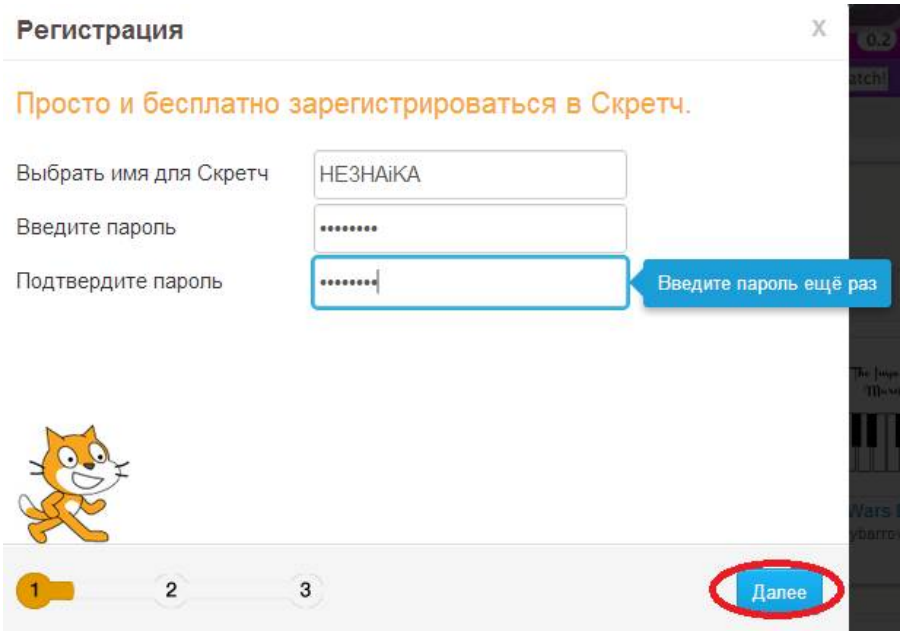
Откроется окно регистрации, в котором в верхнее окно надо ввести Ваше имя для сайта Scratch (иногда это имя называют «логин»). Разработчики советуют не вводить настоящее имя, я тоже не советую.



Придумайте интересный логин, например, типа такого, здесь использованы буквы латинского алфавита и цифра «3» вместо буквы «з».



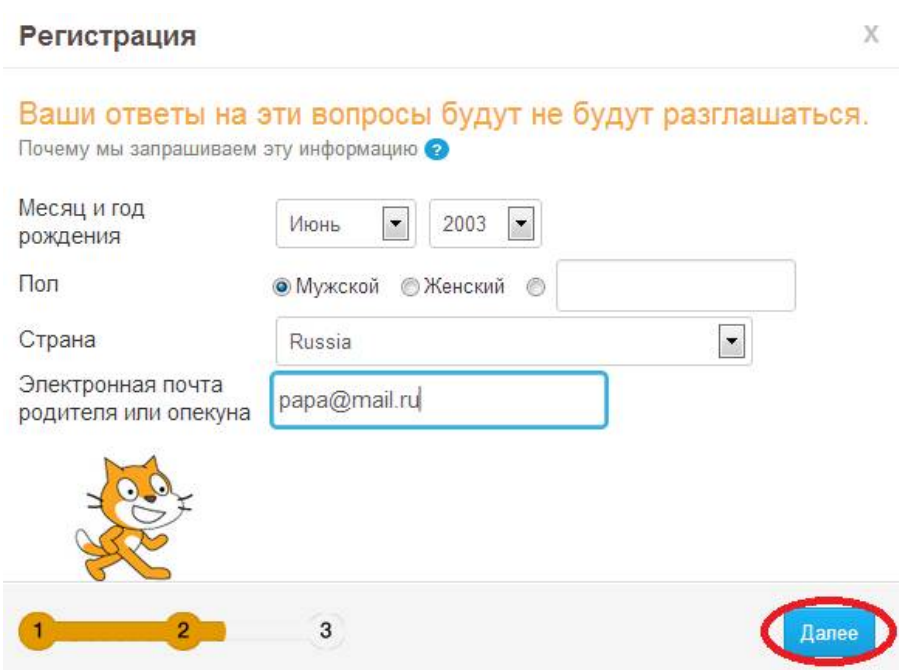
Затем два раза введите пароль и нажмите кнопку «Далее». Не забудьте записать пароль в блокнот!



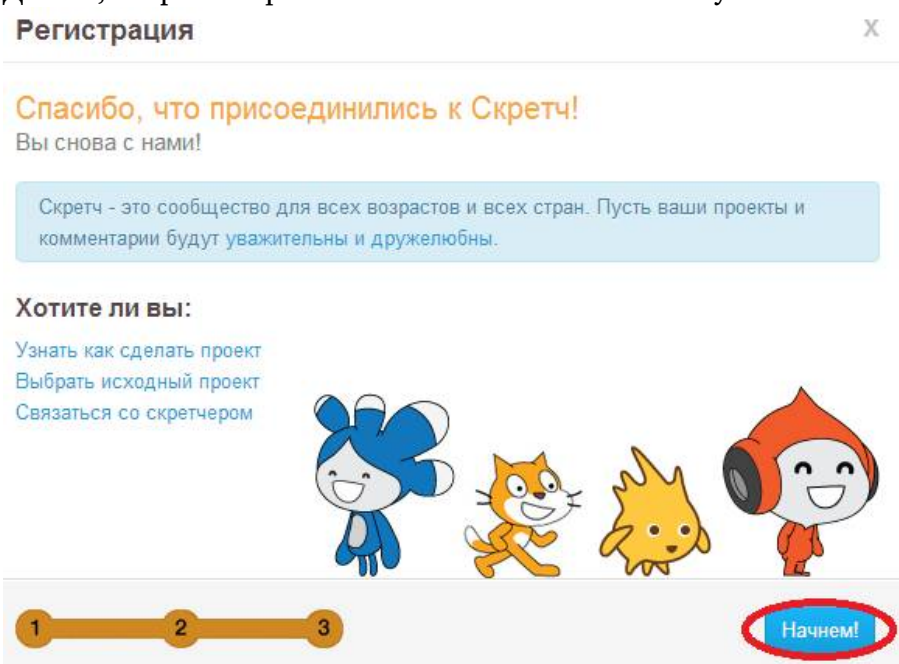
Затем надо заполнить информацию о себе. Здесь лучше написать правду.



Обратите внимание, если Вы прилетели с другой планеты, то кроме мужского и женского пола Вы можете выбрать любой пол и ввести его название, которое принято на Вашей планете.



После того, как Вы заполните информацию о себе и нажмёте «Далее», откроется финальное окно. Нажмите кнопку «Начнём!».



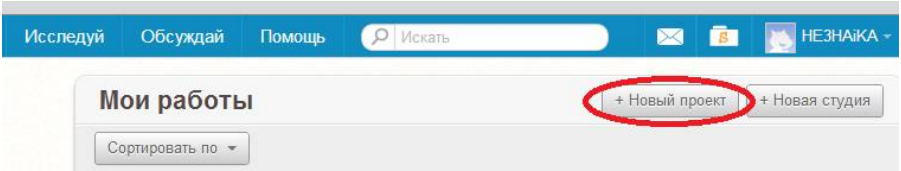
Поздравляю! Вы зарегистрировались! На верху справа Вы увидите Ваш логин и некий конвертик с цифрой один, которая означает, что в почтовом ящике уже есть одно письмо. Кликнув на конвертик, Вы можете прочитать сообщение.



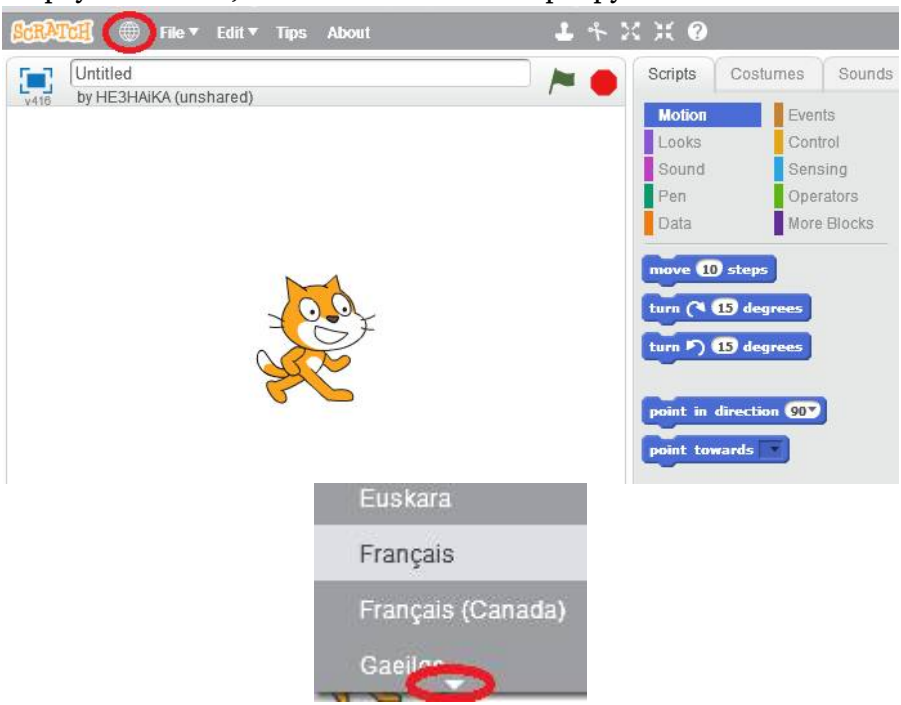
Чтобы начать делать проекты кликните на папку с буквой «S».



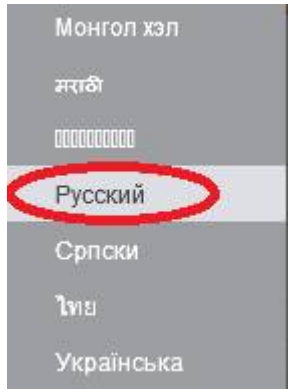
Вы попадёте на страницу «Мои работы». Нажмите на кнопку «+ Новый проект».



Откроется окно Scratch 2. Первым делом надо перейти на русский язык. Нажмите на глобус, двигайте мышку вниз списка, на треугольничек, чтобы список начал прокручиваться.



Почти в самом низу списка будет русский язык.



Готово! Можно начинать создавать проекты.

КАК СКАЧАТЬ И УСТАНОВИТЬ ОФФЛАЙНОВЫЙ РЕДАКТОР SCRATCH.

Это сделать очень просто. Перейдите по адресу http:// scratch.mit.edu/scratch2download/. Если Ваш компьютер работает под управлением Windows, то кликайте туда, куда указывает красный овал.



Затем установите Scratch.



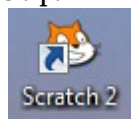
После установки на рабочем столе появится симпатичный ярлык.



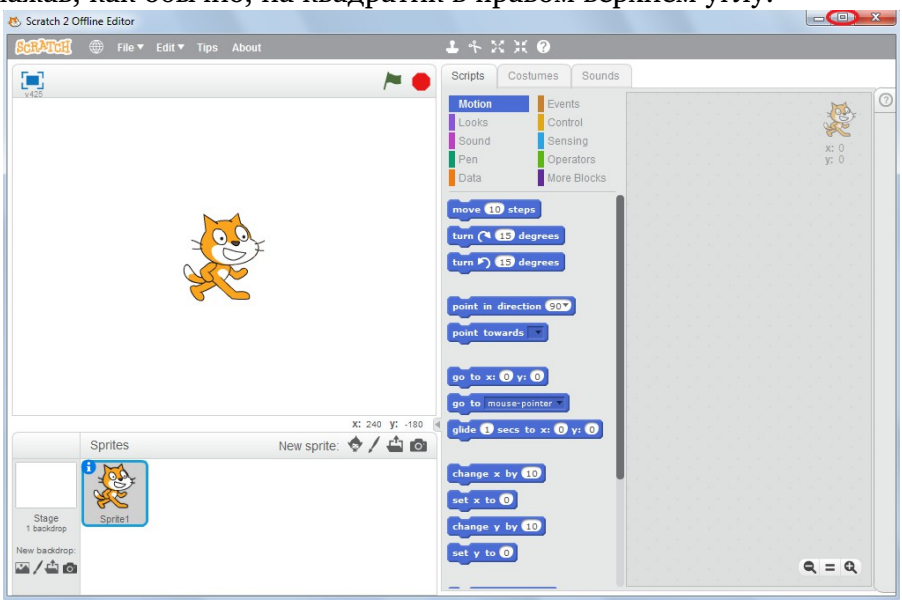
Можно начинать!

ГЛАВА 1. КАК УСТРОЕН SCRATCH. ПЕРВАЯ ПРОГРАММА.

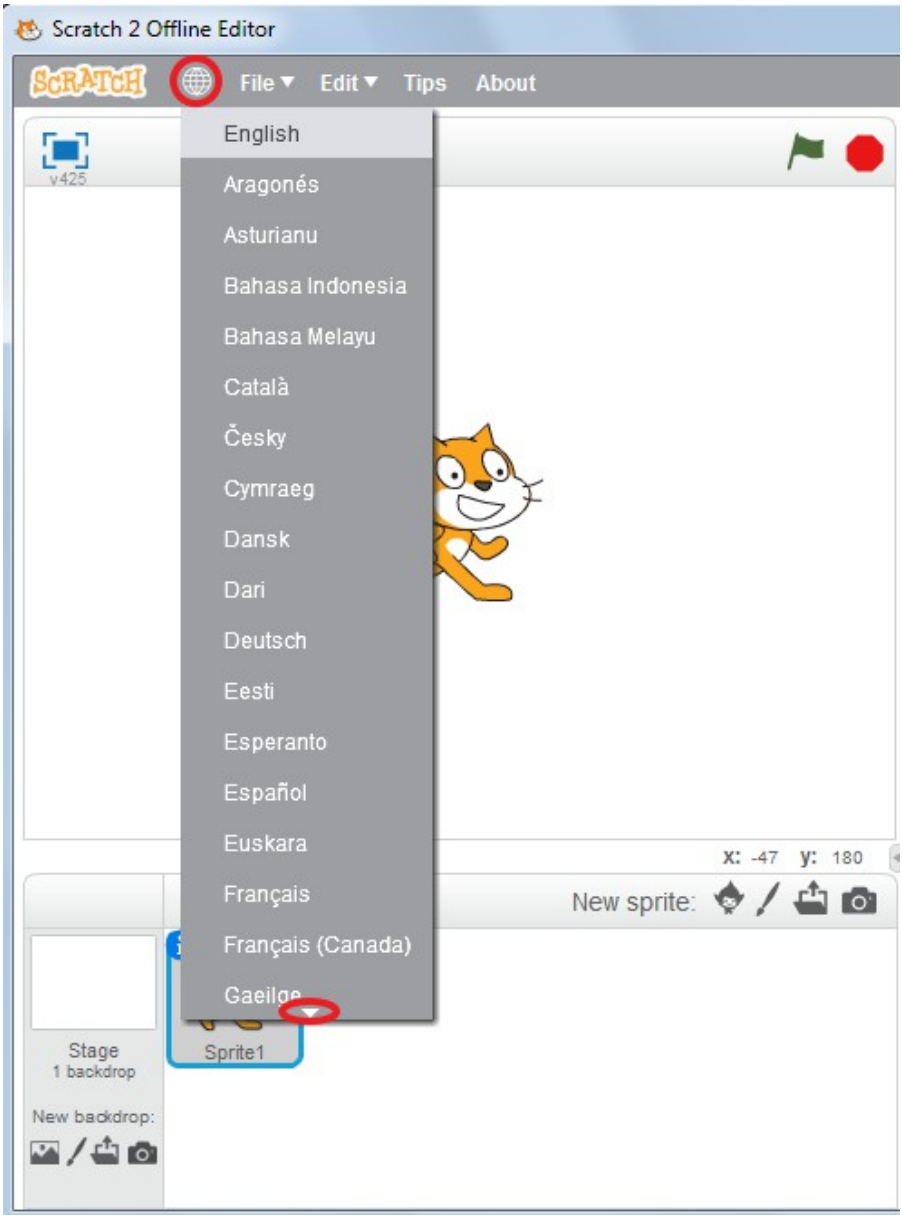
Запустите Scratch 2 через ярлык на рабочем столе.



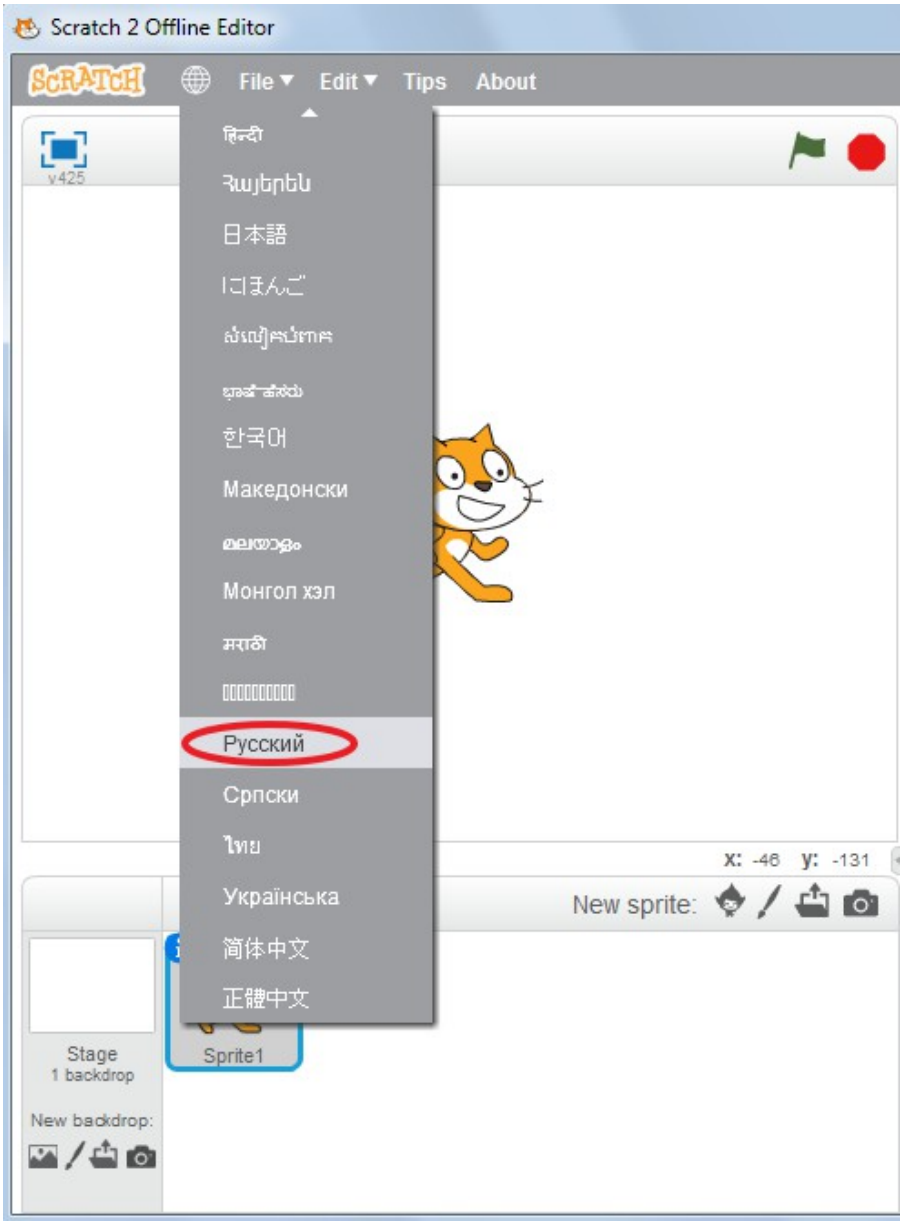
Откроется окно программы. Разверните его на весь экран, нажав, как обычно, на квадратик в правом верхнем углу.



Если программа загрузилась на английском, то первым делом надо переключить программу на русский язык. Кликните на кругленький глобус слева вверху, а затем двигайте мышку на маленький треугольник внизу.

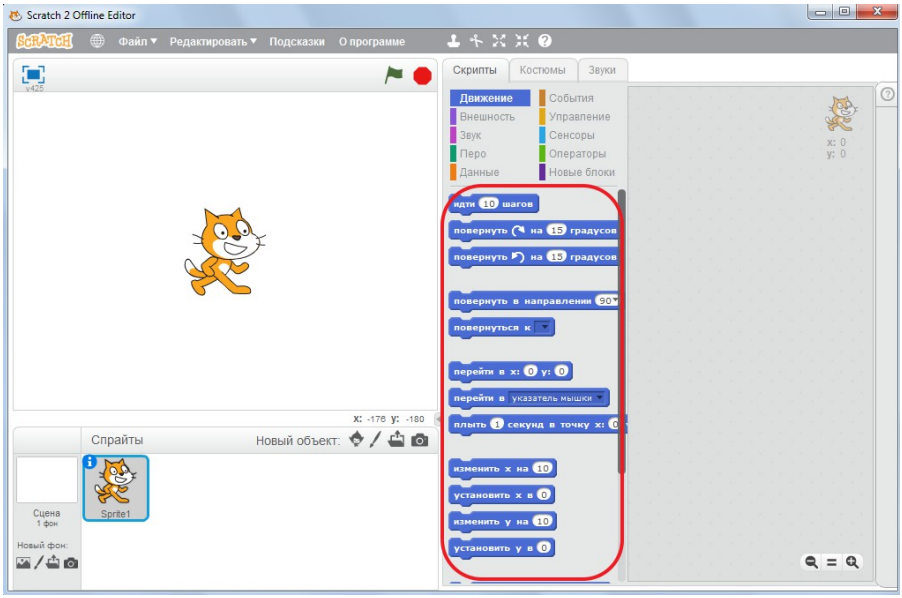


Прокрутите список языков вниз до конца и выберите Ваш родной язык.

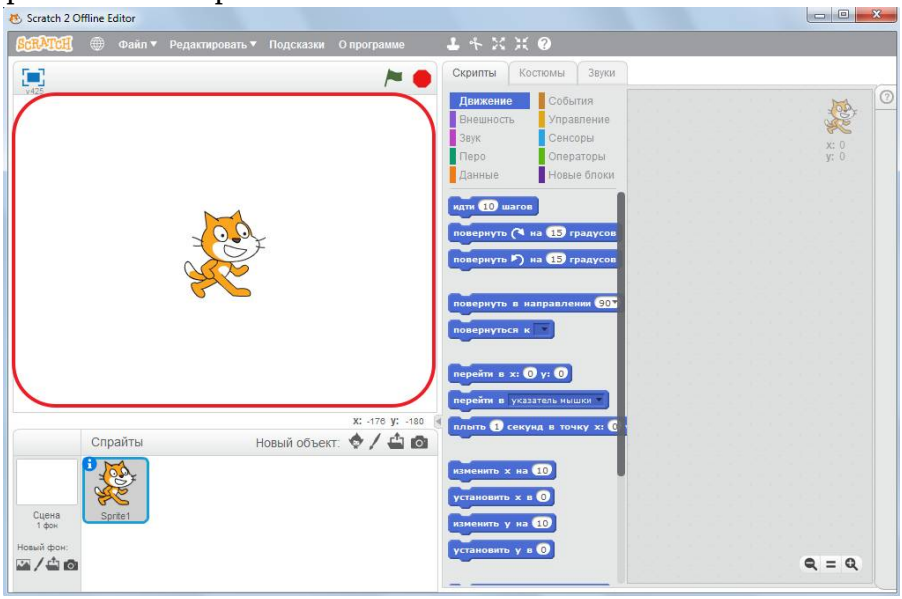


Теперь можно осмотреться, видно, что Scratch 2 состоит из трёх частей.

В центре куча синих блоков.



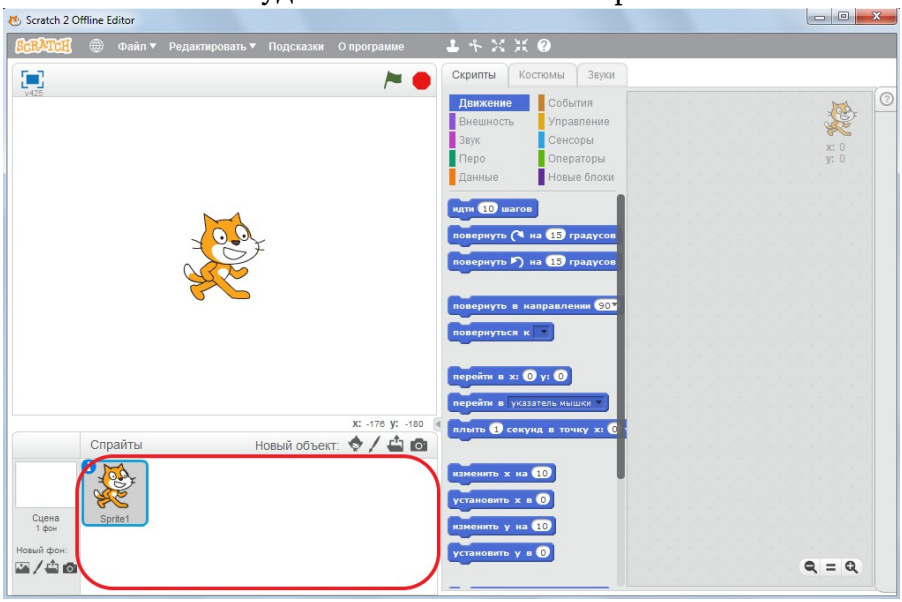
Слева находится Сцена, на которой будет работать программа. На Сцене, как в театре, будут происходить все приключения персонажей.



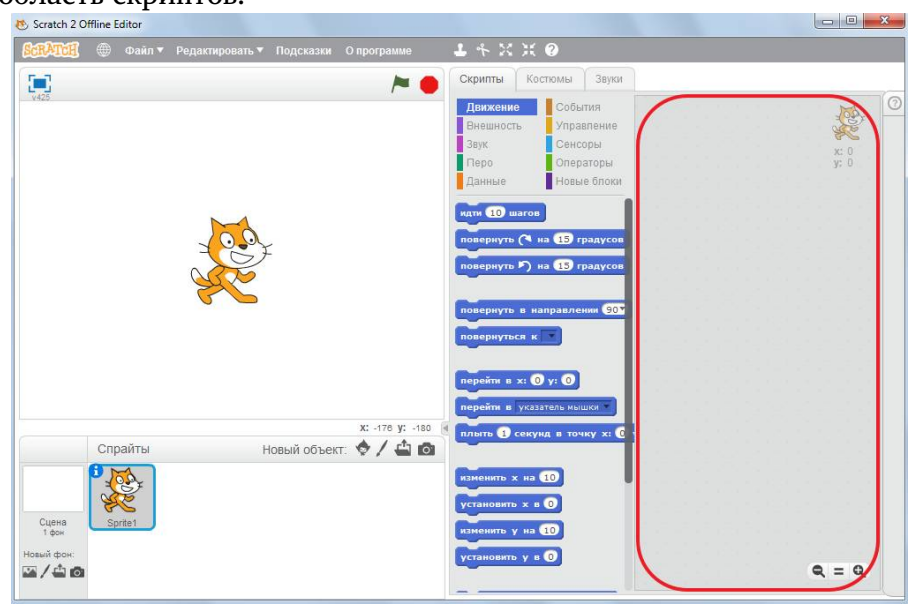
Сейчас у нас только один персонаж — Котик с именем Sprite1. Персонажи в Scratch называются спрайты, наверное, создатели программы пили очень много газировки!



Вот это место будем называть область спрайтов.



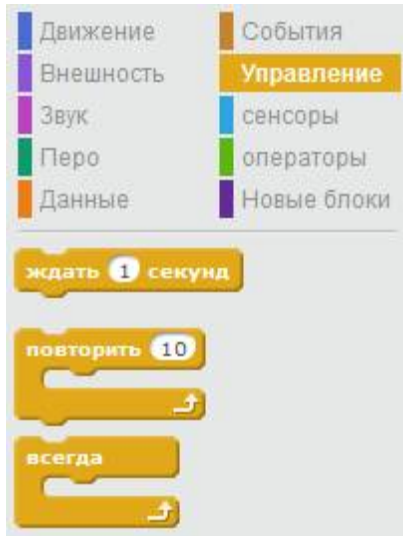
Справа большое серое пустое место, там мы будем складывать программу из блоков, это рабочая область или область скриптов.



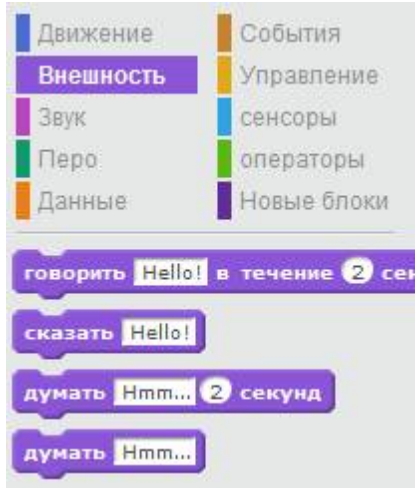
Основа Scratch — разноцветные блоки. Мы в основном будем использовать синие, жёлтые, фиолетовые и голубые. Вот синие блоки.



Вот жёлтые блоки.

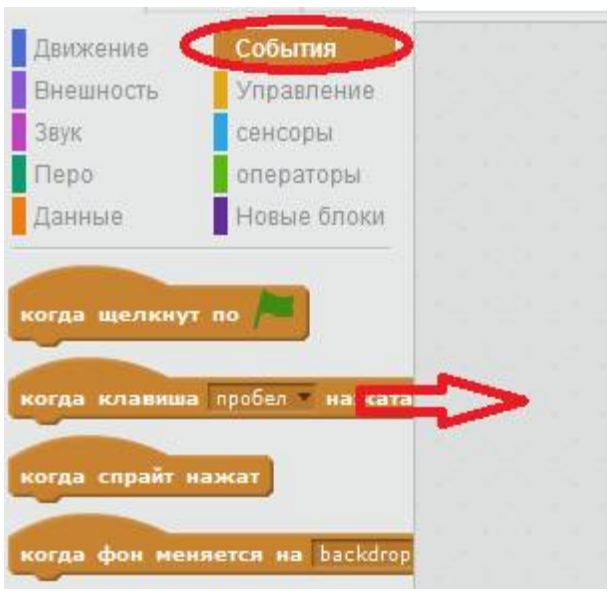


Вот фиолетовые блоки.



Остальные кнопочки Scratch мы изучим в процессе создания программ. А теперь, давайте научим Котёнка бегать.

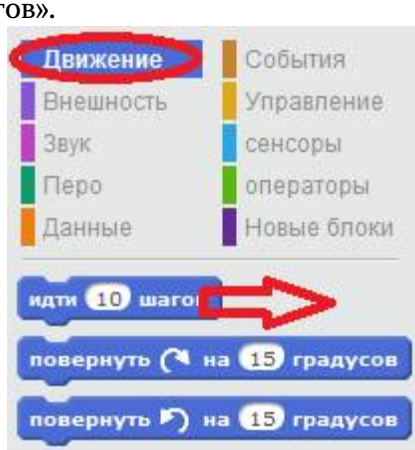
Выбираем вверху блоки «События» и тянем в центр экрана блок «когда клавиша пробел нажата» и там его бросаем. Чтоб тянуть блок, надо нажать на него мышкой и тянуть, а не просто щёлкнуть.



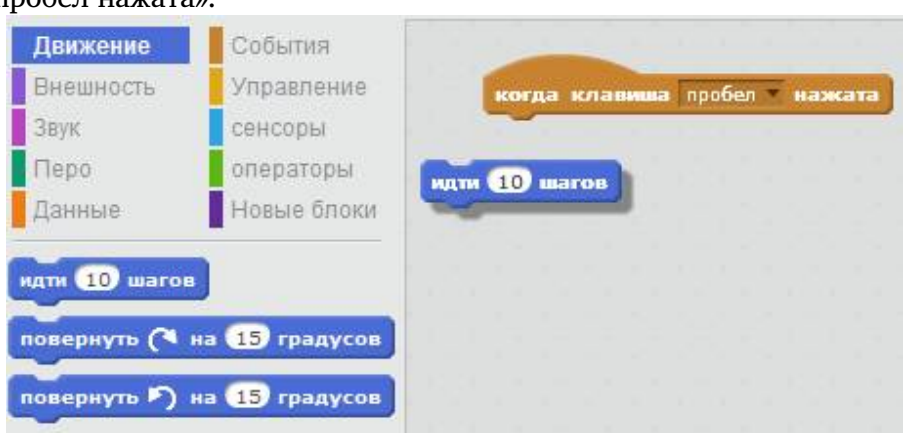
Вытащите блок вправо и бросьте его, отпустив кнопку мышки.



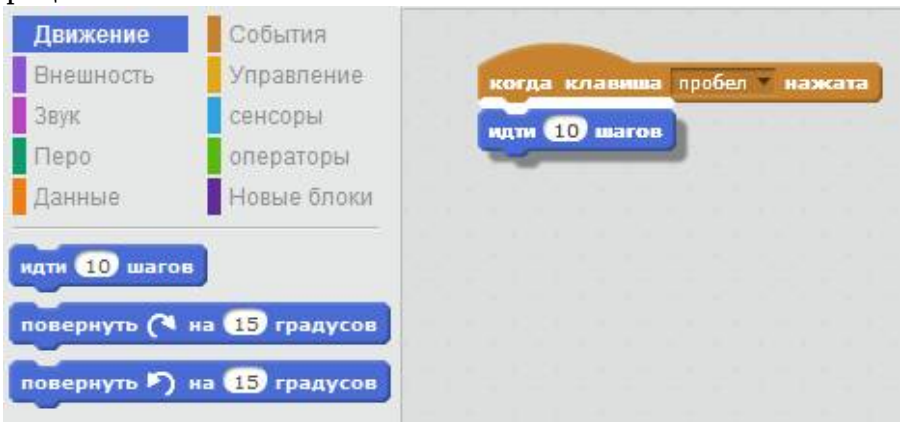
Теперь выбираем синие блоки «движение», тянем направо блок «идти 10 шагов».



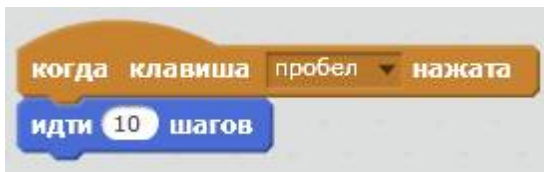
И там его прицепляем под коричневый блок «когда клавиша пробел нажата».



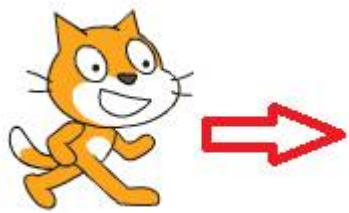
Когда синий блок хочет прицепиться к коричневому — появляется белая полоса. Отпускаем кнопку мышки, и он прицепится.



Ура! Получилась первая программа или скрипт, как её иногда называют.



Нажимайте пробел и посмотрите, как двигается котик.



Если нажимать долго, то он дойдет до края экрана, и почти весь спрячется. Торчит только хвост.



За него-то мы его и вытащим! Жмите на хвост и тащите Кота на центр экрана. Снова нажимайте пробел — Кот опять идет вправо.

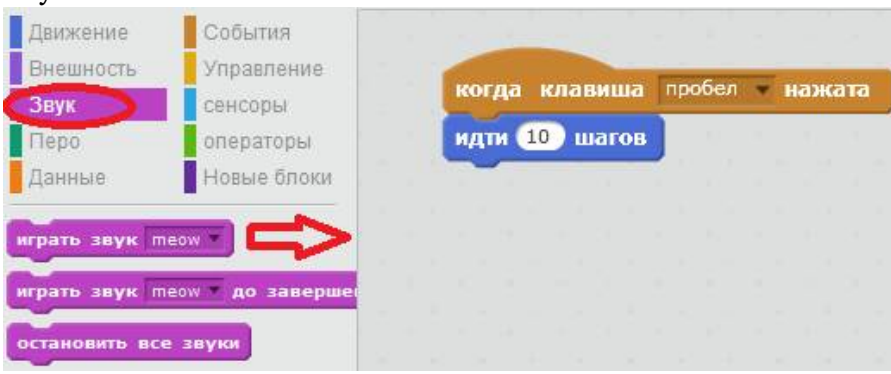
Вытащите его в левую часть экрана и ответьте на вопрос:

сколько раз надо нажать на пробел, чтобы кот дошел до правого края экрана?

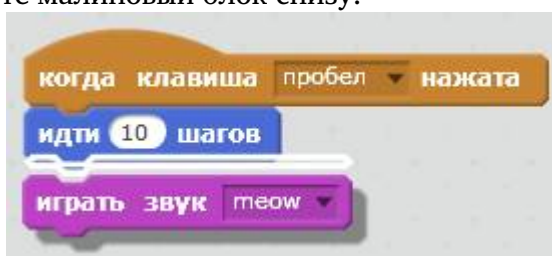


У меня получилось нажать сорок раз. Сорок раз по десять шагов равно четыреста. Получается, что экран примерно четыреста шагов в ширину. Запомните эту цифру.

Теперь добавим в программу звук «мяу». Нажмите на малиновые блоки «звук» и тащите блок «играть звук meow» под синий блок «идти 10 шагов». «Meow» по-английски означает «мяу».

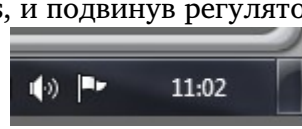


Прицепите малиновый блок снизу.



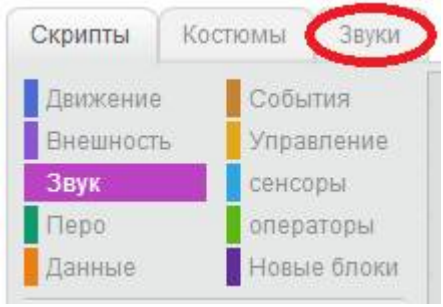
Теперь при нажатии пробела Кот идет и мяукает.

Если он мяукает слишком громко, то отрегулируйте громкость, например, нажав на значок громкости в правом нижнем углу Windows, и подвинув регулятор громкости пониже.

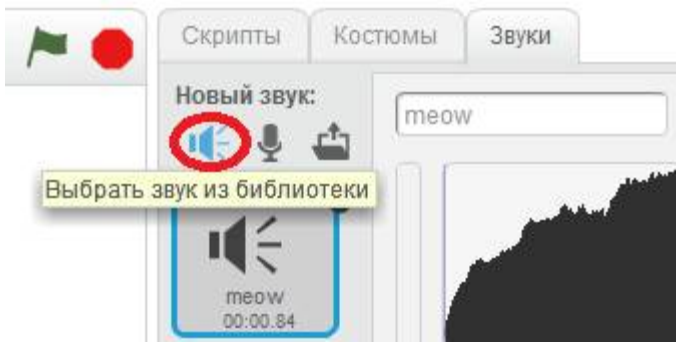


Если мяуканье Вам надоело, то можно добавить коту новых звуков. Пускай, например, он залает!

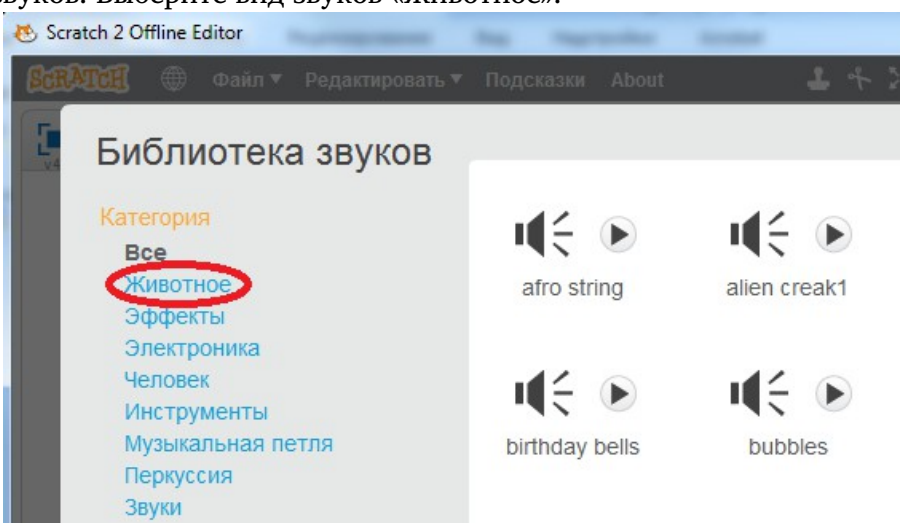
Для этого надо сначала нажать на закладку «звуки».



А потом на кнопку «Выбрать звук из библиотеки».



Откроется Библиотека звуков, в ней есть несколько видов звуков. Выберите вид звуков «Животное».



Все лишние звуки исчезнут, останутся только звуки

животных. Звуки можно прослушать, для этого надо нажать на треугольничек в круге, похожий на кнопку «Play» музыкального плеера.



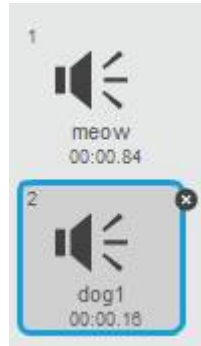
Нажимайте на разные файлы, послушайте звуки животных, представьте, что вы на каникулах в деревне. Потом выберите звук «dog1» и нажмите «ОК».



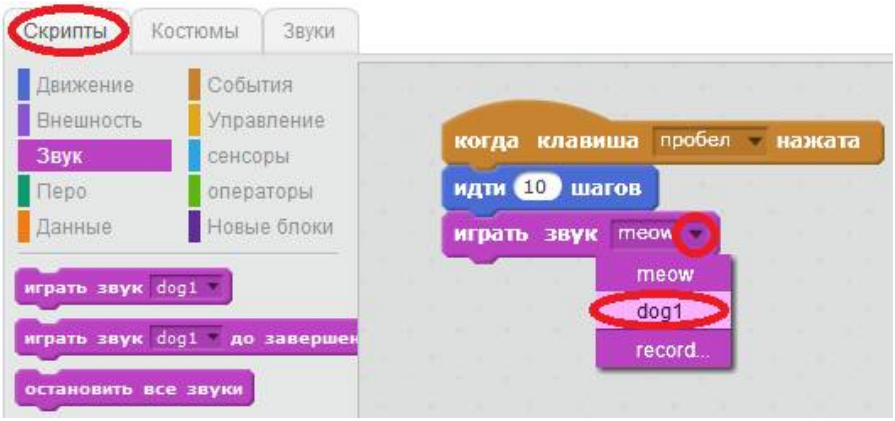
Кнопка ОК расположена внизу.



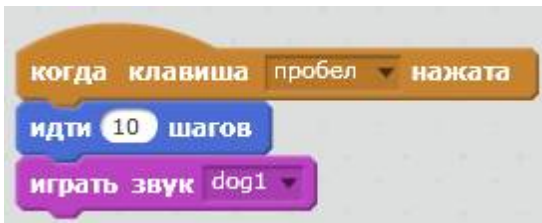
Теперь Кот может использовать в программе два разных звука.



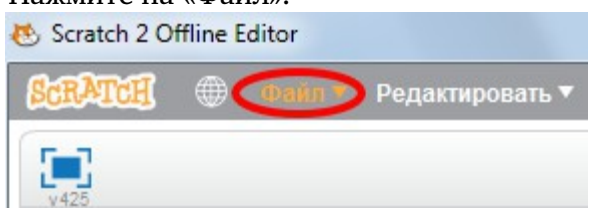
Нажмите на закладку «скрипты» и потом на треугольничек в блоке «играть звук meow», и выберите звук «dog1».



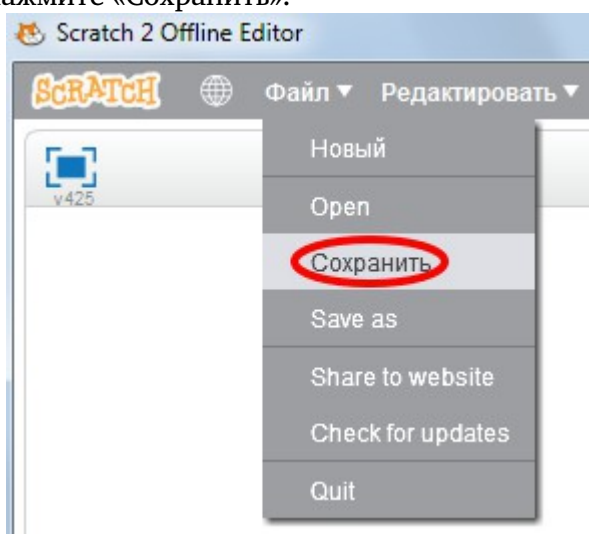
Получилась вот такая программа. Нажимайте пробел, Кот ходит и лает!



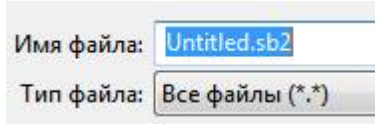
Первая глава подходит к концу, надо сохранить проект, чтобы он не исчез. Нажмите на «Файл».



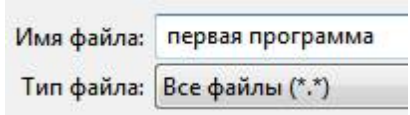
Затем нажмите «Сохранить».



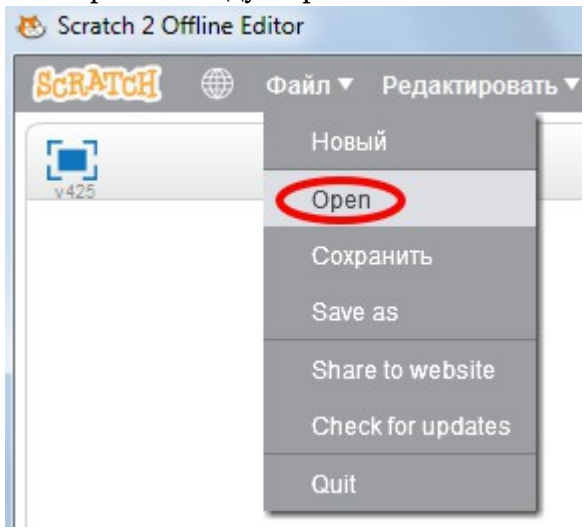
Появится окно сохранения, там нужно ввести имя файла.



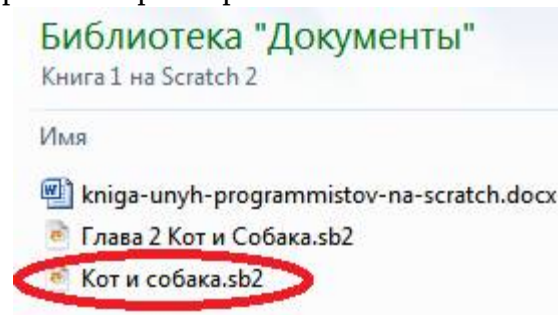
Напишите «первая программа» и нажмите Enter.



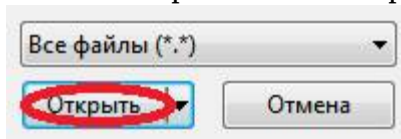
После того, как программа сохранена, её всегда можно запустить снова через команду «Open».



Нужно просто выбрать проект в списке и нажать «Открыть».

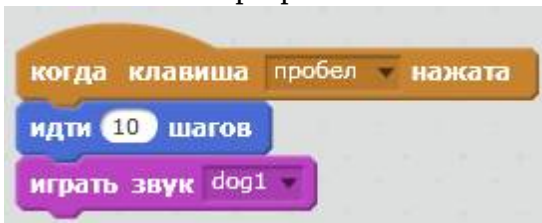


Кнопка «Открыть» обычно расположена справа внизу окна.

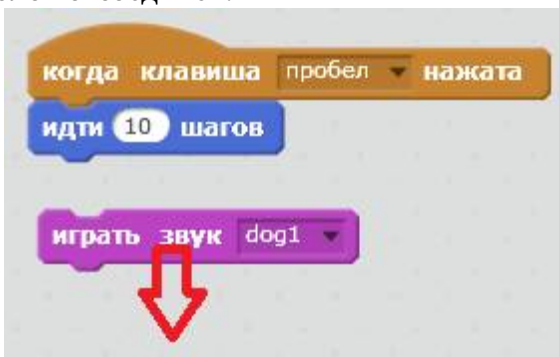


ГЛАВА 2. УСЛОЖНЯЕМ ПЕРВУЮ ПРОГРАММУ.

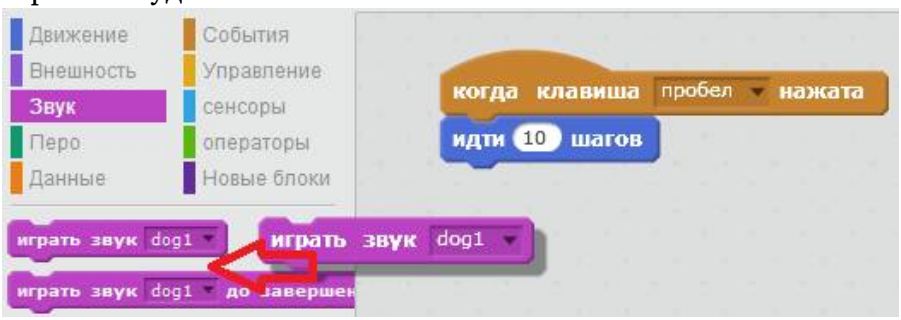
У нас уже есть вот такая программа.



Если Вам наскучил бесконечный лай Кота, то малиновый блок «играть звук dog1» можно удалить. Для этого сначала нажимаем на него и тащим немного вниз и отпускаем. Малиновый блок отсоединён.

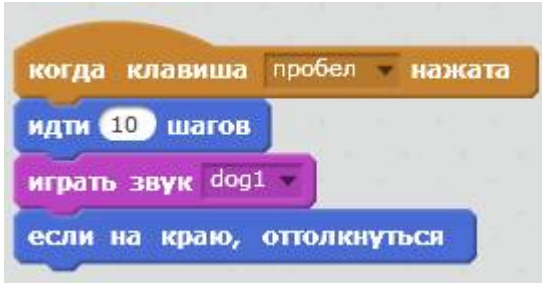


Потом тащим его в левую часть экрана, где лежат все блоки, и бросаем туда. Он исчезнет.



Получится программа для движения Кота без звука. Нажимайте пробел — Кот двигается в тишине.

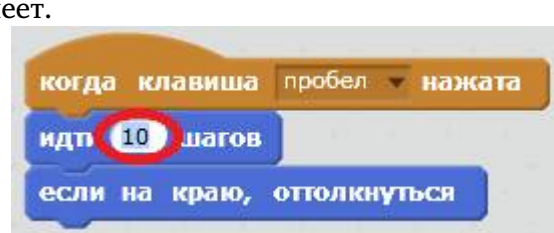
Наверное, Вам уже надоело каждый раз вытаскивать Кота за хвост из-за края экрана. Давайте добавим синий блок «если на краю, оттолкнуться».



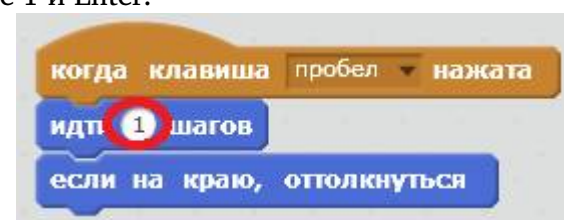
Теперь Кот доходит до края экрана, отталкивается, переворачивается и идёт назад на голове.



Если долго держать пробел, то Кот будет непрерывно бегать влево — вправо и переворачиваться. Попробуйте немного изменить программу. Вместо цифры 10 напишите в блок «идти 10 шагов» цифру 1. Для этого два раза быстро щелкните на цифре 10 — она посинеет.



Нажмите 1 и Enter.



Если теперь держать пробел, то Кот бегает медленнее! Конечно, он же делает за раз всего 1 маленький шаг. Теперь можете побаловаться и вбивать туда разные числа, какие пожелаете, например, пять.

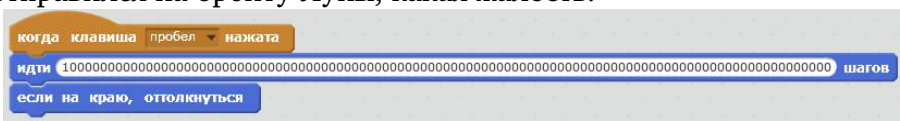


Чем больше число, тем быстрее бегает Котик. Понаблюдайте за его поведением при нажатии пробела.

Один ученик пошутил, и вписал туда 1 и сто нолей!



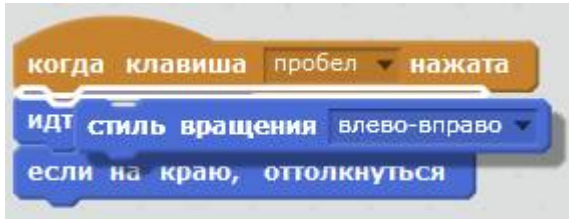
Это число называется Гугол! При нажатии пробела, Кот отправился на орбиту Луны, какая жалость.


когда клавиша пробел нажата
идти 10000000000

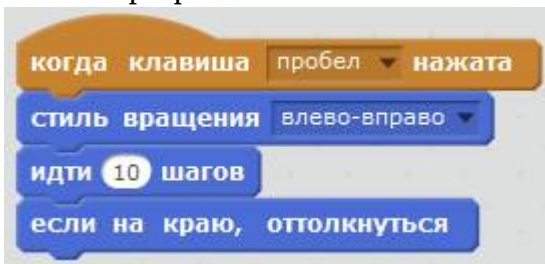
Не обижайте Котика. Не пишите туда числа больше 1000. Мне кажется, что самая нормальная скорость это 10 или 20 шагов.

Теперь давайте сделаем так, чтобы Кот не бегал на голове, ведь это не очень удобно, да и ушки потом болеть будут.

Найдите в самом низу у синих блоков блок «стиль вращения влево-вправо» и тяните его к программе. Положите его прямо под жёлтый блок.



Получится такая программа.

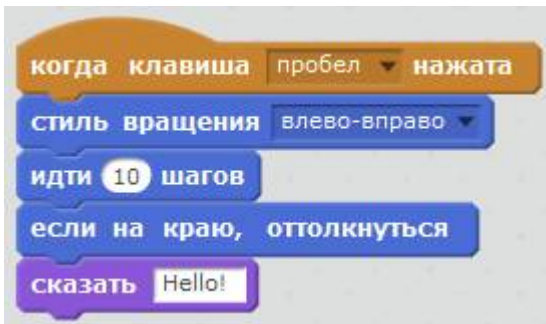


Нажимайте пробел. Кот гуляет как в жизни, всегда ногами вниз.



Сейчас настало время научить Кота разговаривать.

Выберите фиолетовые блоки «внешность» и тащите в программу блок «сказать Hello!».



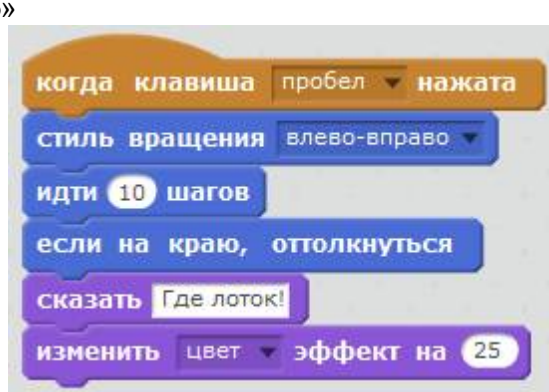
Теперь Кот бегает и на ходу кричит «Hello!».



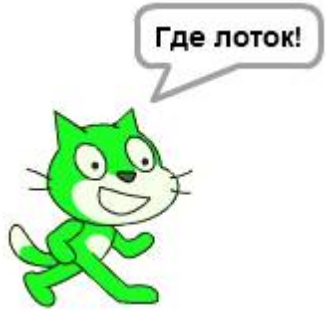
Измените слово «Hello!» на что-нибудь другое. Для этого кликните в текст, и напишите свои слова. Для переключения между русскими и английскими буквами нажмите на клавиатуре сразу две кнопки Shift и Ctrl (на некоторых компьютерах Shift и Alt в левой части клавиатуры).



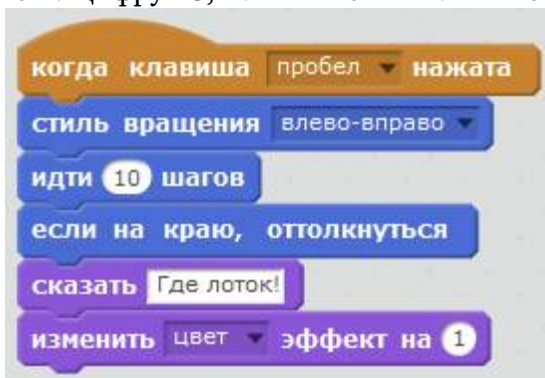
Теперь добавьте снизу фиолетовый блок «изменить цвет эффект на 25»



Кот бегает и меняет цвет!

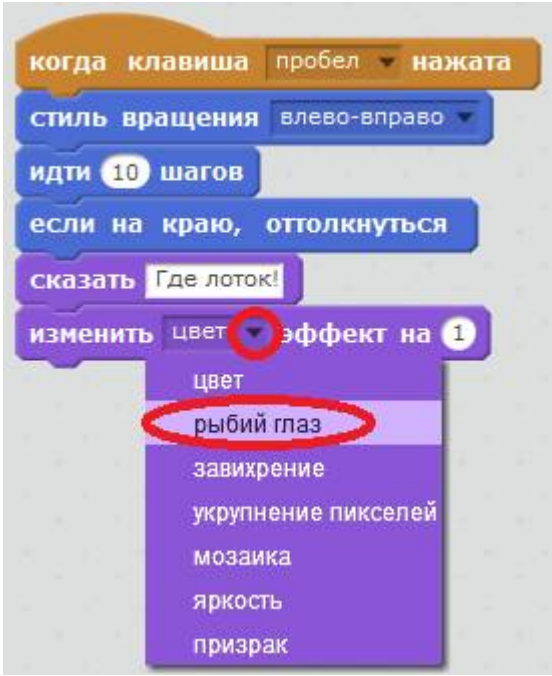


Цвет меняется очень быстро. Измените цифру 25 на 1. Для этого кликните на цифру 25, напишите 1 и нажмите Enter.

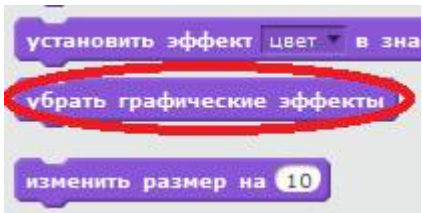


Цвет изменяется плавнее.

Эффект цвет можно поменять на другой эффект. Для этого нажмите на чёрный треугольничек около слова цвет. Выпадет список эффектов.



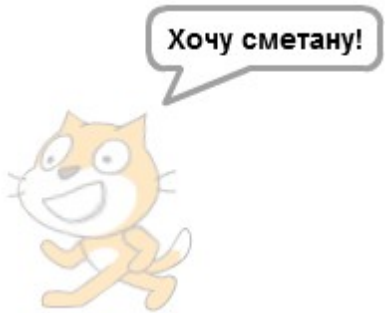
Выберите любой. Теперь можете побаловаться, выбирая разные эффекты и ставя им разные цифры скорости изменения. Больше 25 не ставьте, это не очень интересно. Для того чтобы после применения эффектов возвращать Коту нормальный облик нужно дважды кликнуть на фиолетовом блоке «убрать графические эффекты».



Угадайте, какие два эффекта по очереди применили к Коту?



А сейчас?



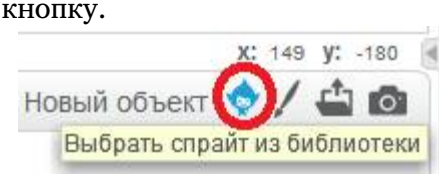
А сейчас?



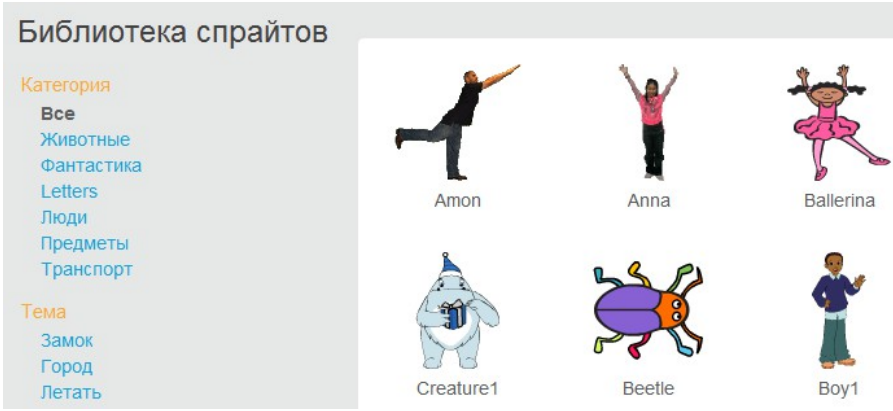
Ну, всё, хватит мучить Кота. Давайте лучше заведем ему друга — добавим в программу нового персонажа. Для этого есть интересная кнопка — «Выбрать спрайт из библиотеки». Если нажать на эту кнопку, то Кот сходит в библиотеку, и принесёт Вам баночку спрайта!



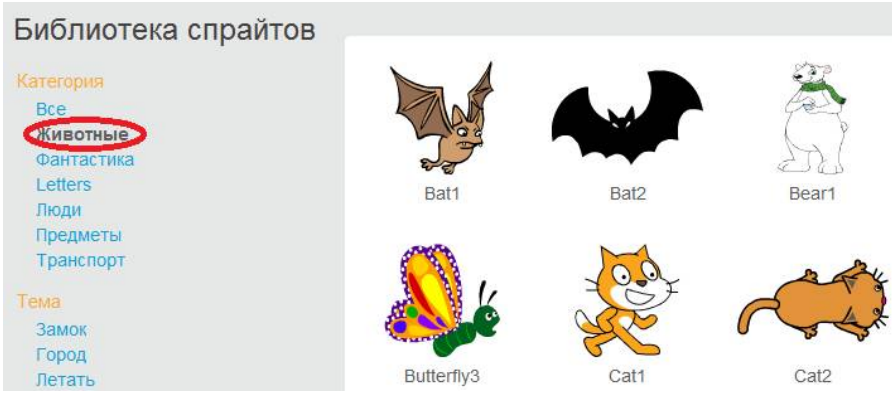
Нажмите на кнопку.



Ха-ха-ха, конечно же, нет! Спрайтами в Scratch называют всех персонажей, которые используются в программе. Сейчас у нас один спрайт — это Кот. Нажмите кнопку «Выбрать спрайт из библиотеки». Откроется Библиотека спрайтов.



Выберите вид спрайтов — «Животные».



Выберите Коту нового друга, например, летучую мышь. Выберите её и нажмите внизу ОК.

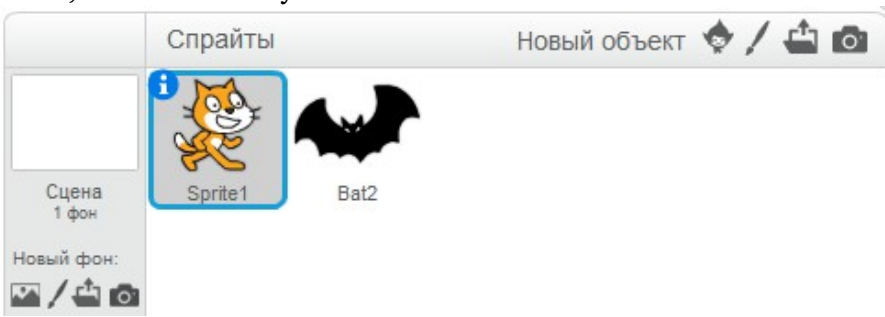


Летучая мышь появится в центре экрана.

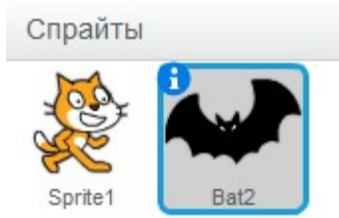


Теперь Коту не скучно!

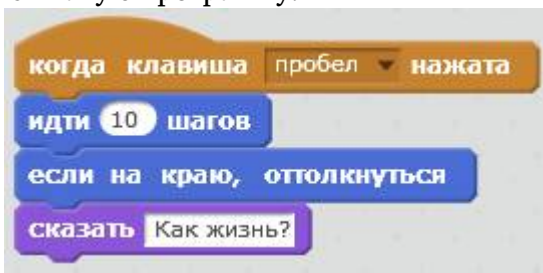
Обратите внимание, слева внизу, в области спрайтов, рядом с Котом, появилась летучая мышь.



Выберите летучую мышь.



Сделайте ей такую программу.



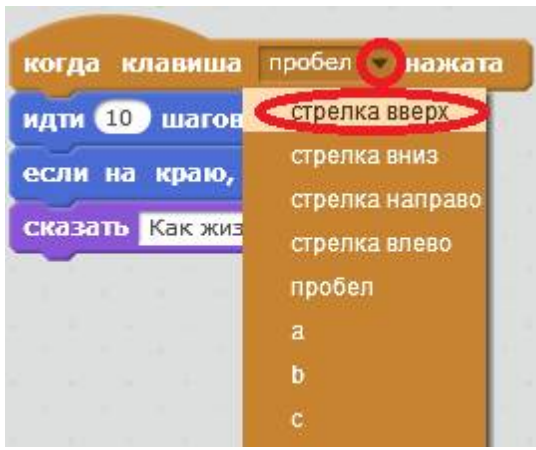
Нажимайте пробел, весёлая компания летает вправо и влево.



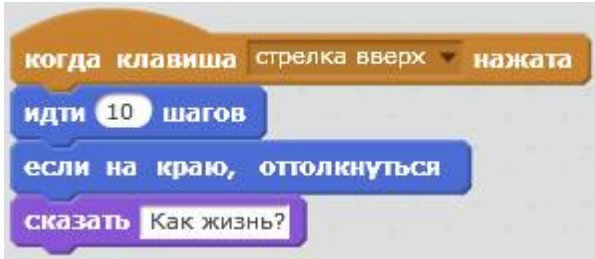
Кот всегда ногами вниз, а мышь иногда переворачивается. Это не страшно, ведь летучие мыши легко могут летать вниз головой.



Сейчас оба персонажа управляются клавишей пробел. Давайте сделаем их независимыми. Кот по-прежнему будет управляться пробелом, а летучая мышь будет управляться стрелкой вверх. В программе летучей мыши нажмите на чёрный треугольничек, выпадет список всех клавиш, выберите стрелку вверх.



Теперь программа летучей мыши выглядит так.



Нажимайте стрелку вверх — двигается только летучая мышь. Нажимайте пробел — двигается только Кот. Попробуйте изменить его управляющую клавишу. Выберите Кота в области спрайтов. Нажмите на чёрный треугольничек и выберите любую клавишу. Если Вы выбираете английскую букву, то не забудьте переключить клавиатуру на английский язык, а то клавиша работать не будет.

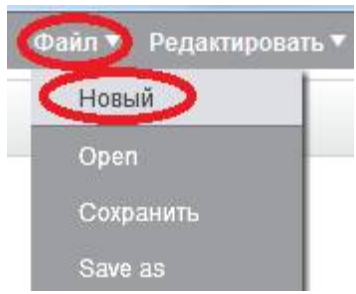


Не забывайте, что для переключения между русскими и английскими буквами надо нажать сразу две кнопки Shift и Ctrl (на некоторых компьютерах Shift и Alt в левой части клавиатуры).

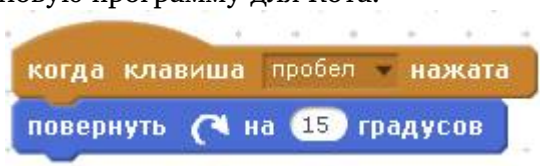
Сохраните получившийся проект.

ГЛАВА 3. ЦИКЛИЧЕСКОЕ ВЫПОЛНЕНИЕ ПРОГРАММЫ.

Циклическое — повторяющееся по кругу, когда одно и то же действие выполняется снова и снова. Создайте новый проект. Нажмите «Файл», потом «Новый».



Сложите новую программу для Кота.



Нажимайте на пробел, Котик будет поворачиваться на месте.



Совет. Для того, чтобы развернуть Кота вправо дважды кликните блок «повернуть в направлении 90».

Давайте сделаем так, чтобы при нажатии на пробел Кот совершал полный оборот. Сначала посчитайте сколько раз надо нажать на пробел, чтобы он снова встал вертикально. Получается двадцать четыре раза. Правильно, всего в круге 360 градусов. Если 360 разделить на 15, то получится 24 раза. Для того, чтобы Котик при нажатии пробеле совершил один оборот, надо добавить в скрипт блок «повторить». Выберите блок «повторить» и тащите его на существующий скрипт. Когда блок «повторить» обнимет блок поворота, то отпустите его.



Получится вот такой скрипт.



Нажмите пробел. Котик сделает один оборот вокруг своей оси. Несколько раз переверните Котика и запомните, с какой скоростью он вращается. Теперь немного измените скрипт. Сделайте так, чтобы он повернулся 36 раз по 10 градусов.



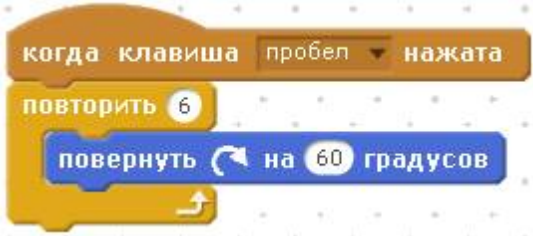
Нажмите на пробел — Котик крутится медленнее! Ещё немного измените скрипт.



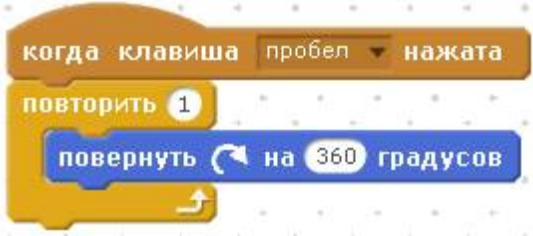
Кот крутится совсем медленно. Думаю, вы теперь сами догадаетесь, как ускорить вращение Кота. Надо увеличить количество градусов, и уменьшить число повторов так, чтобы произведение осталось 360. Например, так.



Протестируйте скрипт, нажав на пробел. Вот это скорость! Давайте ещё ускорим.



А теперь давайте сделаем максимальное ускорение! Один оборот сразу на 360 градусов!

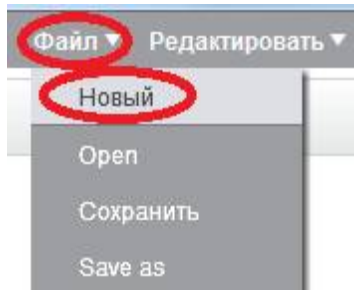


Нажмите на пробел. Не работает? Ещё как работает! Только переворот настолько быстрый, что вы его не замечаете! Шутка! На самом деле блок «повторить 1 раз» не имеет смысла, ведь и без этого каждый блок Scratch выполняется по одному разу. Также, как и поворот на 360 градусов не имеет смысла, ведь положение спрайта никак не измениться.



Запомните, и никогда не делайте таких глупых скриптов!

Давайте теперь сделаем минимультик про Кота и Собаку с использованием новых блоков. Создайте новый проект.

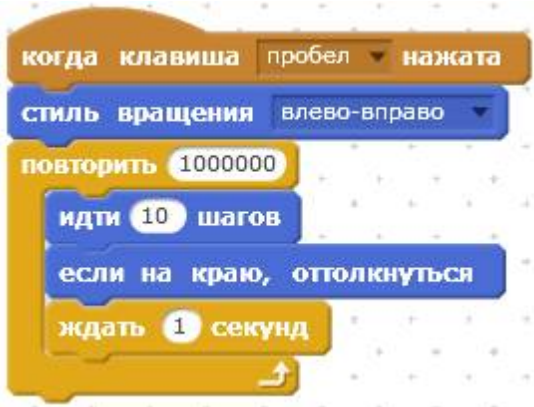


Сделайте такую программу для Кота.



Синие блоки Вам уже знакомы. Появились два новых жёлтых блока. При выполнении блока «ждать 1 секунду» выполнение программы останавливается на одну секунду, и ничего не происходит. Блок «повторить 10» обнимает два синих блока и один жёлтый. Блоки оказываются вложенными в жёлтый блок повтора, они будут выполняться поочерёдно десять раз подряд. Нажмите пробел, Кот делает 10 раз по 10 шагов и останавливается. Сколько всего шагов проходит Кот? 10 * 10 = 100 шагов. Звёздочка * это знак умножения. Нажмите на пробел и подождите десять секунд пока Кот не нагуляется. Немного измените программу. Напишите в блоке «повторить» число двадцать. Теперь Кот гуляет двадцать секунд. Напишите там шестьдесят. Теперь Кот гуляет целую минуту.

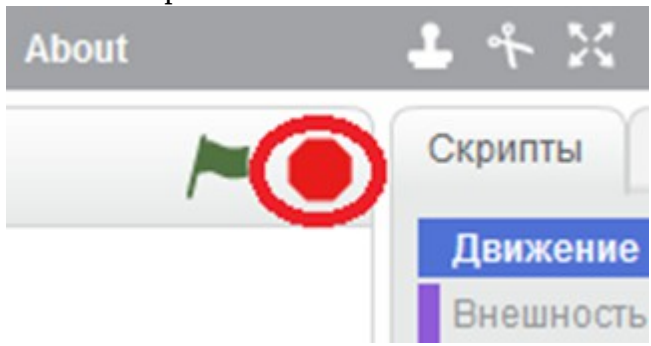
Если же в блоке «повторить» написать цифру 1000 (тысяча) или 1000000 (миллион), то программа будет повторяться очень долго.



Кот будет бегать без остановки миллион секунд!



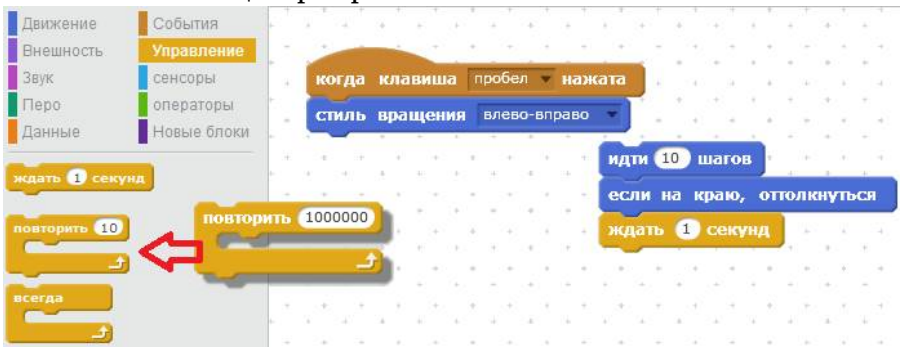
Это примерно 11 лет. Те, кто не хочет ждать, могут остановить бег кота красной кнопкой.



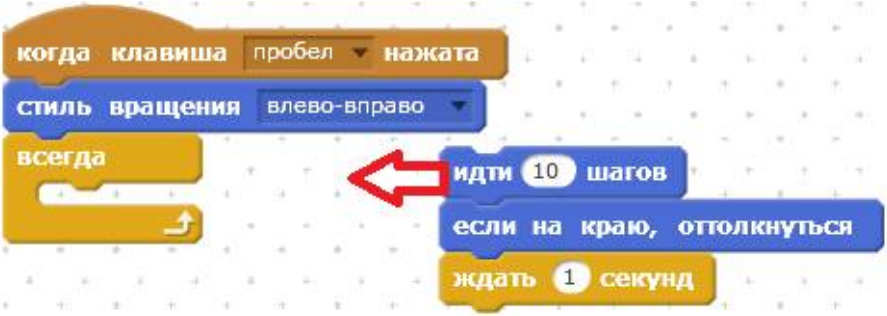
Очень большие числа в блок повторения лучше не писать, это не правильно. Правильно — использовать блок «всегда». Давайте посмотрим, как он работает. Заменим блок «повторить 1000000» на блок «всегда». Для этого сначала вынем из блока «повторить 1000000» три вложенных блока. Возьмите мышкой за верхний вложенный блок «идти 10 шагов», и вытащите три блока на любое пустое место.



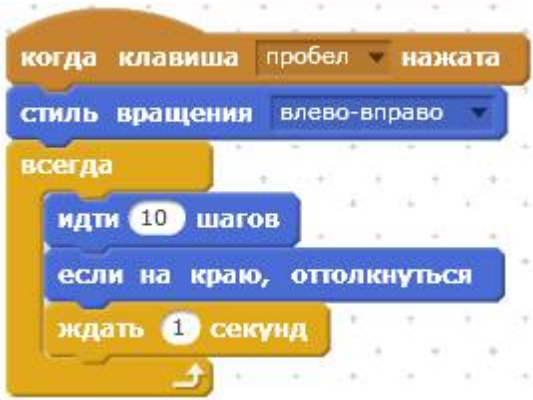
Затем отцепите блок «повторить 1000000» и перетащите его ко всем блокам в центр экрана. Он исчезнет.



Вместо исчезнувшего блока «повторить 1000000» возьмите блок «всегда» и прицепите к блоку «стиль вращения влево-вправо».



Затем затащите в него три блока.

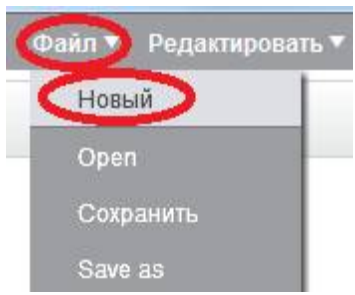


Теперь один раз нажмите Пробел. Кот шагает без остановки. Так как мы использовали блок «всегда», Кот будет гулять, пока мы его не остановим красной кнопкой.



Блок «всегда» мы будем очень часто использовать в наших проектах.

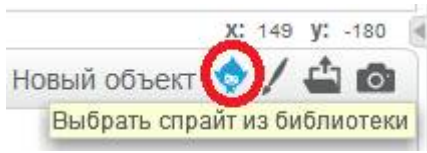
С помощью блока «всегда» можно сделать наш первый интерактивный проект. Интерактивный, это такой, в котором можно управлять движением персонажей. Создайте новый проект, так же, как мы делали в начале главы.



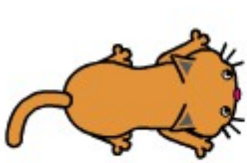
Старый проект можно не сохранять.



Появился новый проект, в котором есть только один спрайт — Кот. Добавьте еще один спрайт, нажав на кнопку «Выбрать спрайт из библиотеки».



Добавьте кого-нибудь из животных, например, Собаку.


Cat2


Crab


Dan


Dog Puppy




Dog2

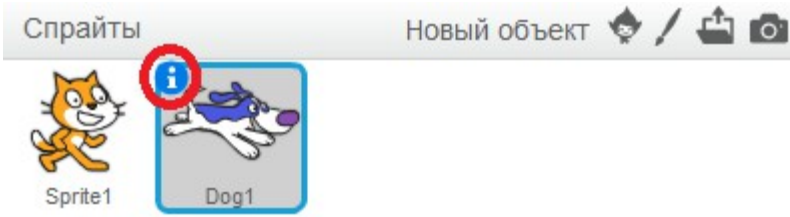
Видите, Собака появилась прямо над Котом.



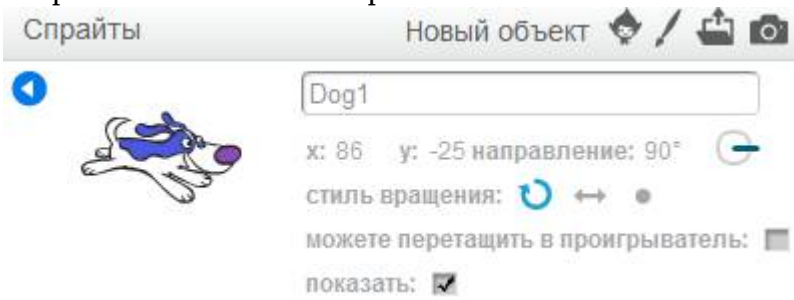
Растащите их. Кот — влево, Собака — вправо.



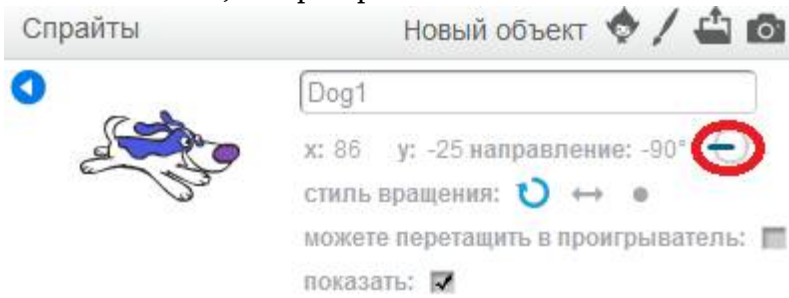
Получилось неплохо, но, обычно Собака бежит не от Кота, а к нему! Собаку надо развернуть мордочкой к Коту. Повернуть Собаку можно в свойствах спрайта. Нажмите на синий кружочек с буквой «i» у спрайта Собаки.



Откроется окно свойств спрайта.



Нажмите мышкой на синюю палочку, которая показывает направление Собаки, и переверните её влево.

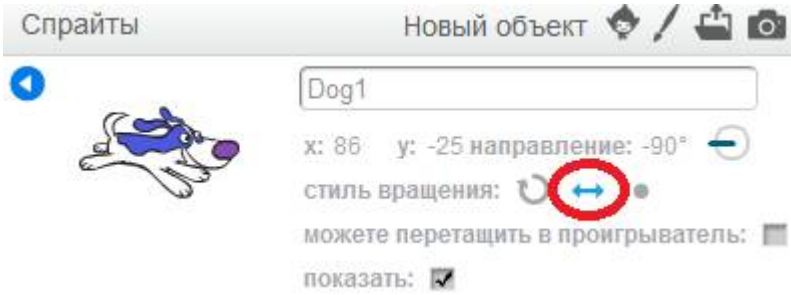


Собака перевернулась вверх ногами!





Да уж, лёжа на спине она Кота не догонит. Для того чтобы собака не переворачивалась на спину, надо нажать маленькую кнопку с двумя стрелочками.



Совсем другое дело!

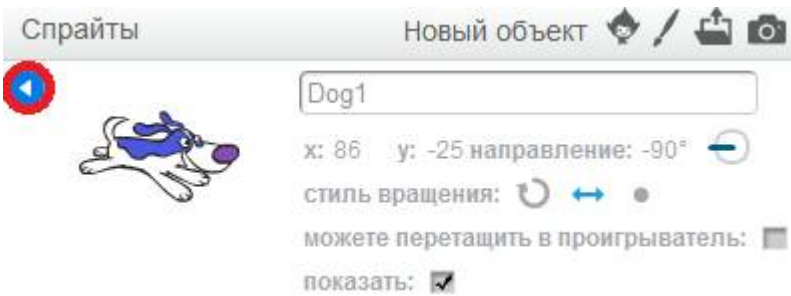


«Привет, Котяра!»
«Привет, привет. Кто краску синюю на тебя пролил?»

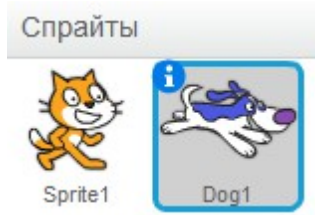




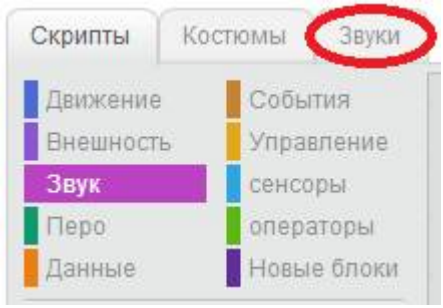
Закройте окно свойств спрайта нажав на синий кружочек с треугольником.



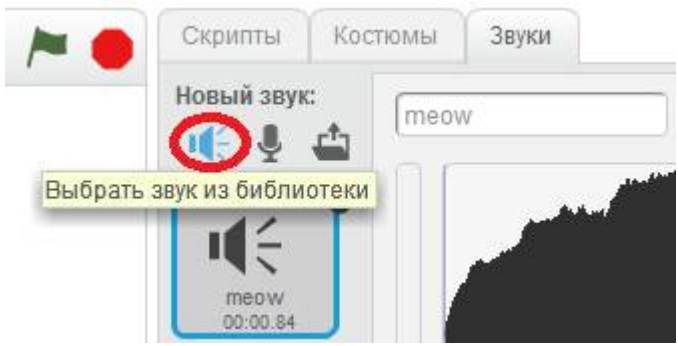
Кот в нашем проекте будет мяукать, а Собака будет лаять. У Кота звук «мяу» добавлен всегда, а у собаки пока ещё нет никаких звуков. Надо добавить ей звук «dog1». Убедитесь, что выбран спрайт Собаки.



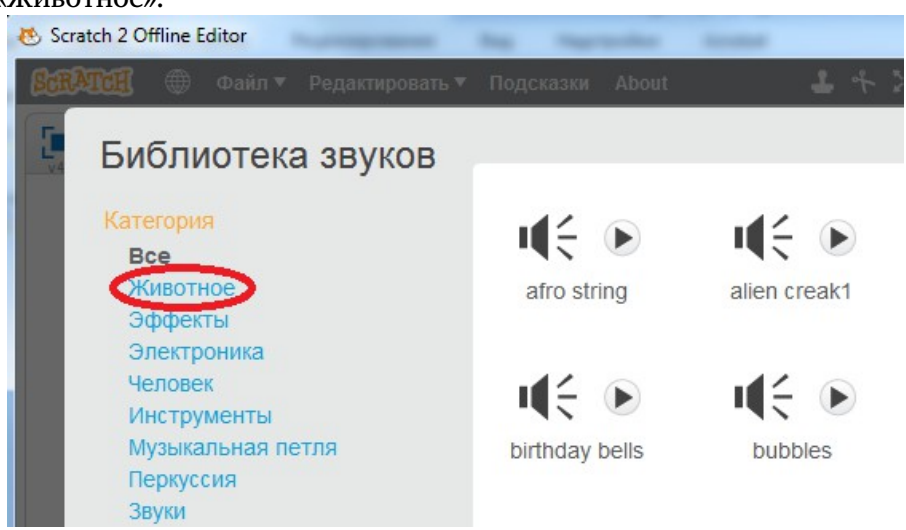
Как добавлять звуки, Вы уже знаете, для этого надо сначала нажать на закладку «звуки».



А потом на кнопку «Выбрать звук из библиотеки».



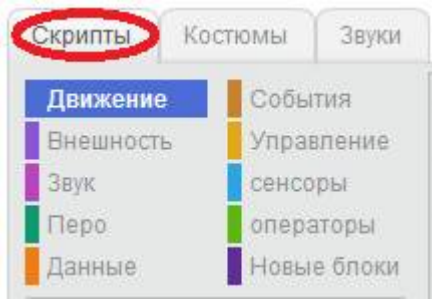
Откроется Библиотека звуков. Выберите вид звуков «Животное».



Все лишние звуки исчезнут, останутся только звуки животных. Выберите звук «dog1» и нажмите «ОК».



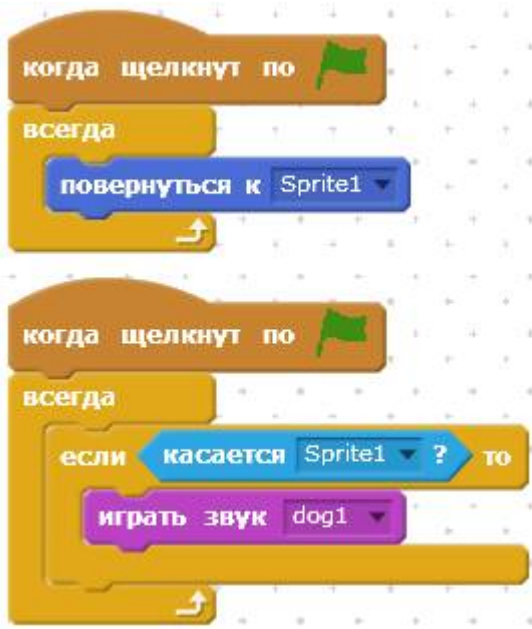
Теперь у нас есть два спрайта со звуками. Можно начинать программировать! Перейдите на вкладку «Скрипты».



Убедитесь, что выбран спрайт Собаки.



Сделайте ей такую программу из двух скриптов. В этих скриптах использованы новые блоки. Синий блок «повернуться к Sprite1», голубой блок «касается Sprite1» и жёлтый блок «если».

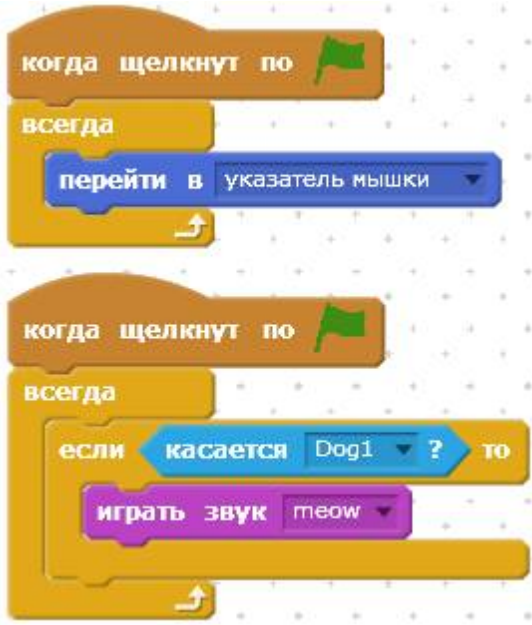


Программа состоит из двух скриптов, которые будут работать одновременно. При нажатии на зелёный флажок Собака всегда будет повёрнута в сторону Кота, за это отвечает верхний скрипт. При касании Кота собака будет лаять, за это отвечает второй скрипт.

Теперь запрограммируем Кота, выберите его в области спрайтов.



Для Кота сделайте такую программу. В ней тоже два скрипта.



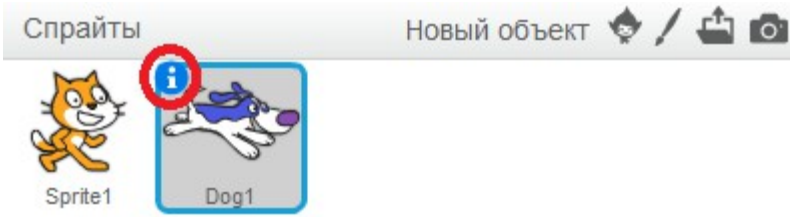
Запустите программу, нажав на зелёный флажок.



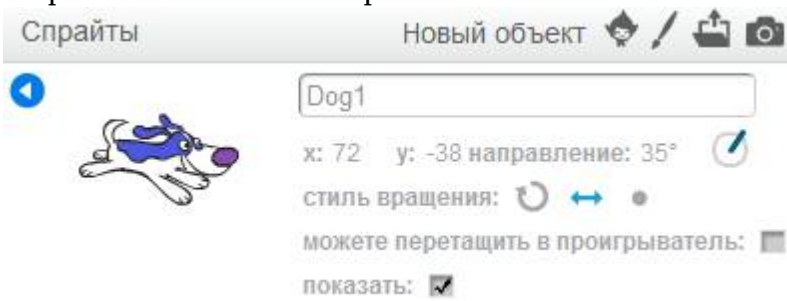
Кот прыгнет к курсору мышки и приклеится к нему. Это происходит из-за блока «всегда перейти в указатель мышки». Кот будет следовать за мышкой до тех пор, пока Вы не остановите программу красным кружком. Попробуйте аккуратно приблизиться Котом к Собаке. Как только они соприкоснуться, раздастся лай и мяуканье, точно, как в жизни, когда Вы бросаете кота на соседскую собаку.



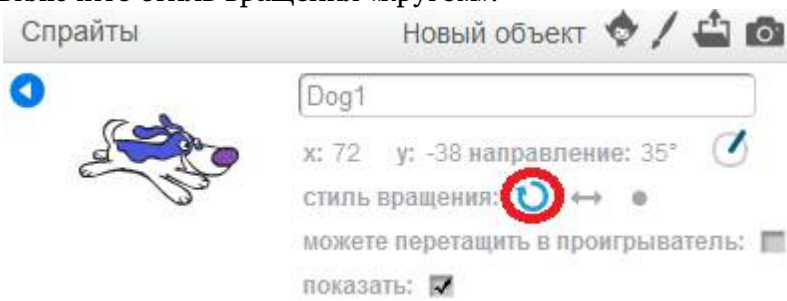
Если Собаки не касаться, а просто прогуливаться котом где-нибудь вверху экрана, то собака будет поворачиваться к Коту, не спуская с него глаз. Как Вы можете заметить, Собака поворачивается только влево или вправо. Это происходит из-за того, что в свойствах спрайта «Dog1» включён стиль вращения «влево-вправо». Давайте изменим режим поворота, чтобы Собака всегда смотрела прямо на Кота. Остановите работу программы красным кругом. Откройте свойства спрайта «Dog1».



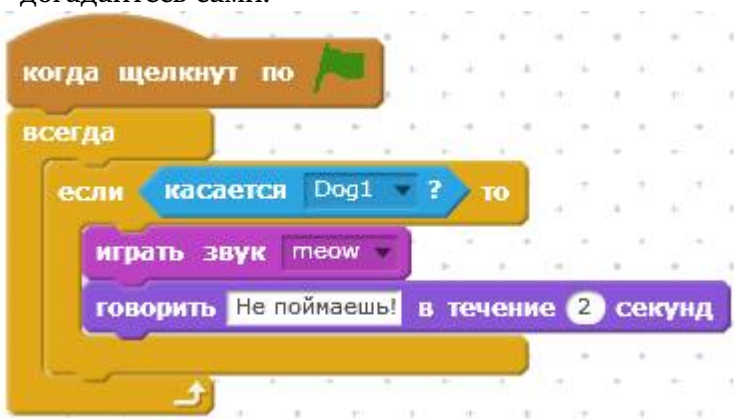
Откроется окно свойств спрайта.

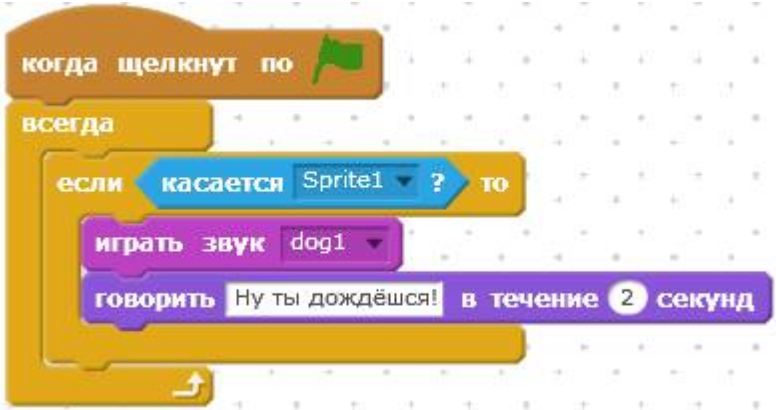


Включите стиль вращения «кругом».



Запустите программу зелёным флажком. Подвигайте Кота — теперь Собака всегда смотрит прямо на него. Чтобы стало интереснее, можно добавить персонажам слова. Какую фразу кому — догадайтесь сами.

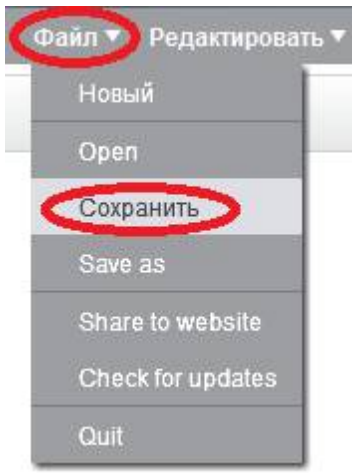




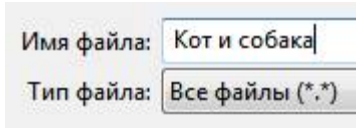
В этом проекте Собака так и не догнала Кота. Когда Вы научитесь программировать лучше, сможете это исправить.



Программа получилась интересная. Давайте сохраним её на память. Потом сможем показать друзьям и родителям. Нажмите «Файл», потом «Сохранить».

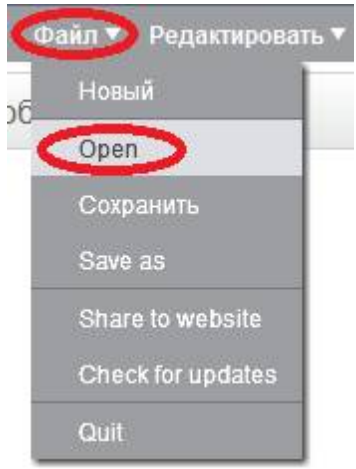


Откроется окно сохранения. Выберите папку, в которую будете сохранять проекты, и напишите имя проекта, например, «Кот и собака» и нажмите Enter.

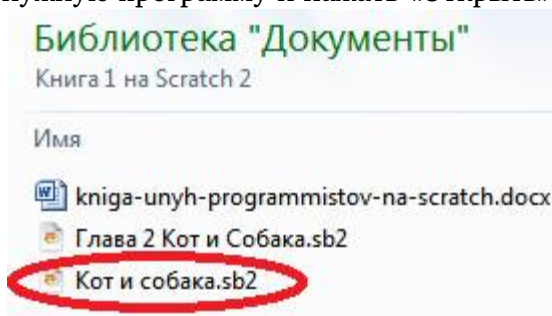


Теперь программа никуда не пропадет.

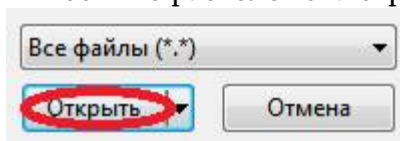
Для того чтобы показать программу друзьям, после включения Scratch надо нажать «Файл», а потом «Open».



Выбрать нужную программу и нажать «Открыть».



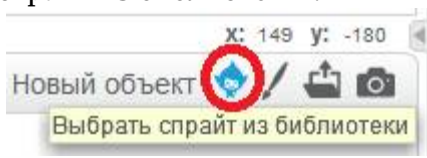
Кнопка «Открыть» обычно расположена справа внизу окна.



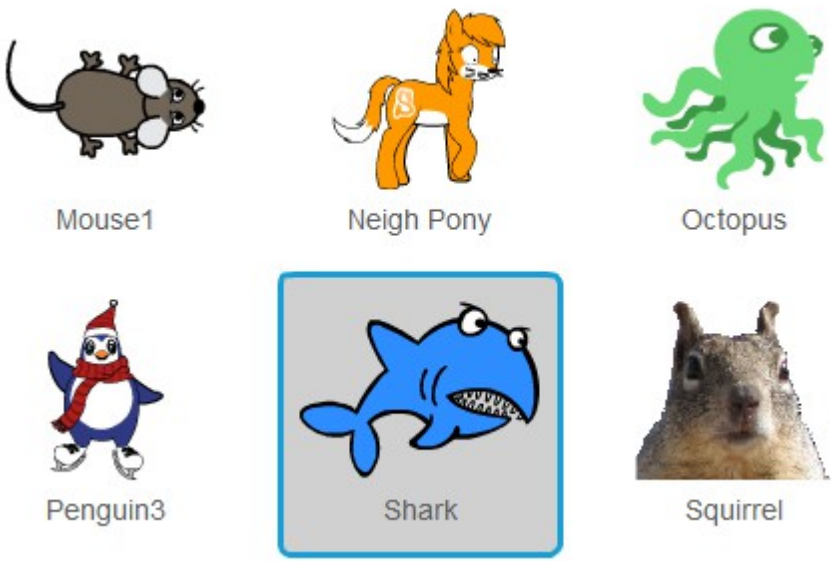
ГЛАВА 4. ПЕРВЫЙ ПРОСТЕНЬКИЙ МУЛЬТФИЛЬМ.

Предыдущий проект был интерактивным мультфильмом. Мы могли управлять одним из персонажей. Теперь давайте сделаем мультфильм без управления персонажами.

Запустите Scratch. Добавьте два новых спрайта, нажав на кнопку «Выбрать спрайт из библиотеки».



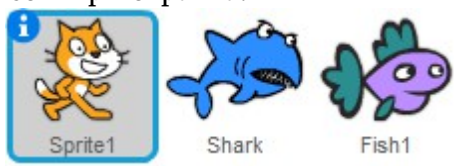
Выберите Акулу и нажмите ОК.



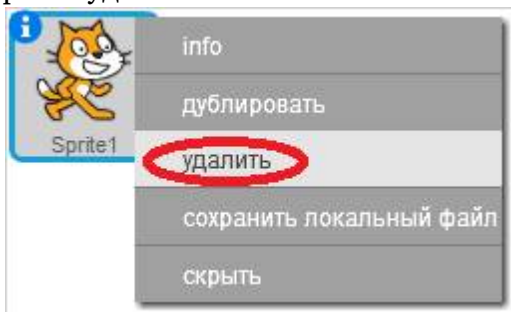
Затем выберите Рыбку и нажмите ОК.



Теперь у нас есть три спрайта.



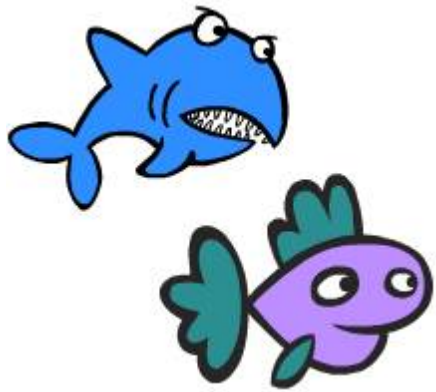
Кота в нашем мультфильме не будет, удалите его через меню правой кнопки. Для этого кликните в Кота правой кнопкой мышки, и выберите «удалить».



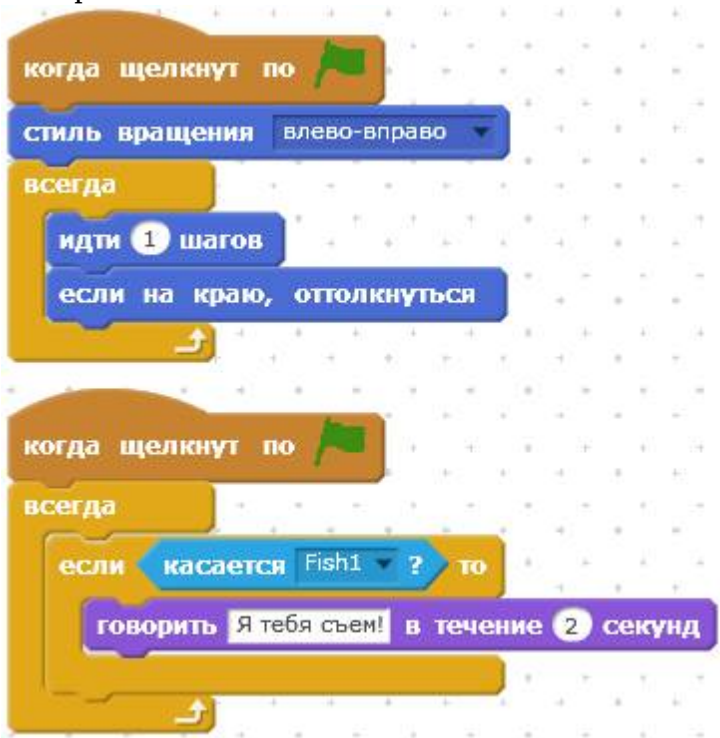
Остались только водоплавающие.



Расположите Акулу повыше, а Рыбку пониже.



Рыбка пока будет стоять на месте, а Акула будет плавать и приговаривать: «Я тебя съем!» — в момент касания Рыбки. Сначала сделаем программу для Акулы. Обратите внимание, программа Акулы состоит из двух частей (из двух скриптов), оба этих скрипта запускаются при нажатии на зелёный флажок и работают одновременно. Верхний скрипт отвечает за движение Акулы, а второй за касание Рыбки.



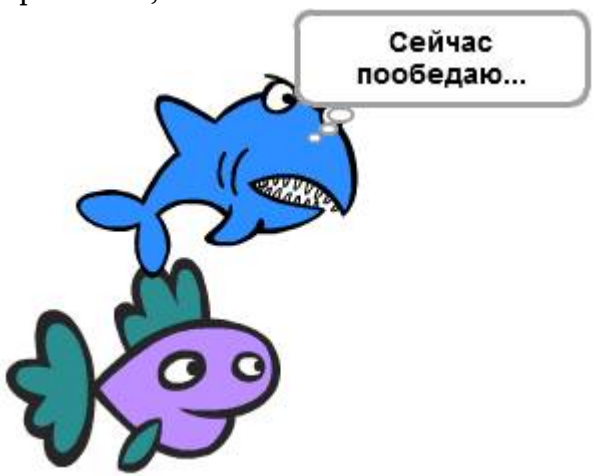
Ой! Кажется, рыбы не разговаривают…



Но думать то они умеют! Давайте заменим блок «говорить» на блок «думать». Второй скрипт примет такой вид.



Нажмите зелёный флажок, Акула поплывёт. Делает это она очень медленно, по одному шагу за раз. Если Акула плавает и ничего не говорит, значит, она не касается Рыбки. Немного подвиньте персонажей, чтобы они касались.



Надо дать Рыбке шанс на спасение. Пусть она тоже плавает. Скопируйте ей скрипт с движением Акулы. Нажмите на верхний блок скрипта движения, тащите его на Рыбку, и отпустите в тот момент, когда курсор мышки будет точно над спрайтом Рыбки.



Теперь выберите спрайт Рыбки.



У неё появился такой же скрипт движения, как и у Акулы.



Запускаем программу. Красота! Все плавают! Но Рыбка никак не реагирует, когда её кусает Акула. Давайте скопируем ей и разговорный скрипт от Акулы. Снова выберите спрайт Акулы.



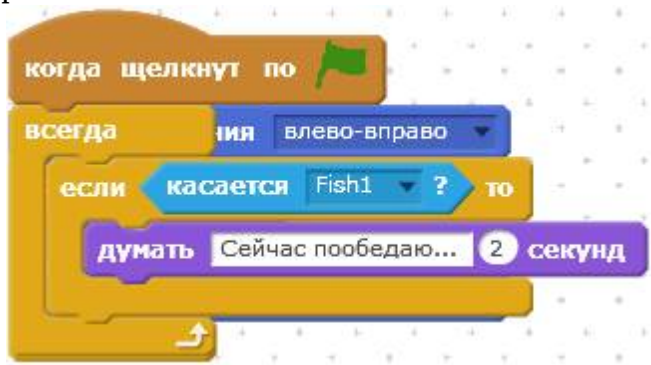
Тащите второй скрипт на Рыбку и отпустите в тот момент, когда курсор мышки будет точно над спрайтом Рыбки.



Теперь выберите спрайт Рыбки.



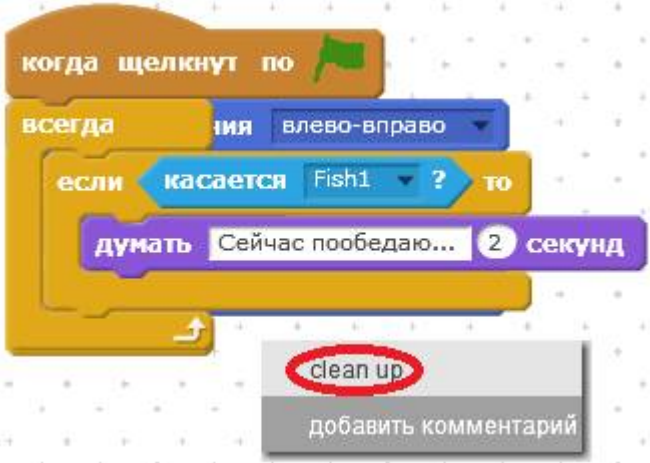
Кажется, скрипты смешались в кучу, второй скрипт лежит поверх первого!



Для того чтобы привести расположение скриптов в порядок, есть хитрый способ.



Кликните правой кнопкой мышки в пустое пространство, и выберите «clean up».



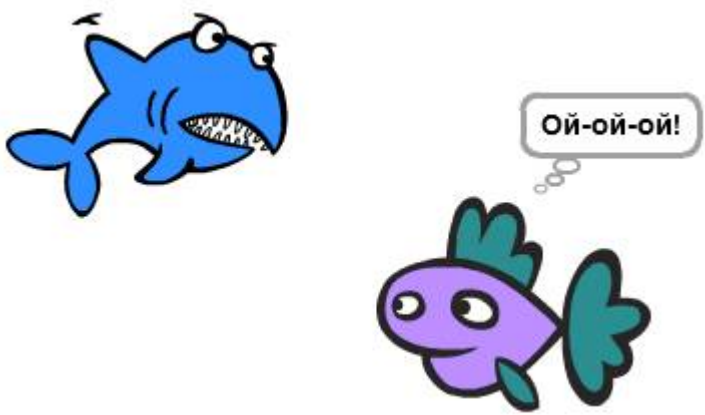
Блоки расположились в ряд. Расположение блоков не влияет на работу программы. Можете расположить их вертикально.



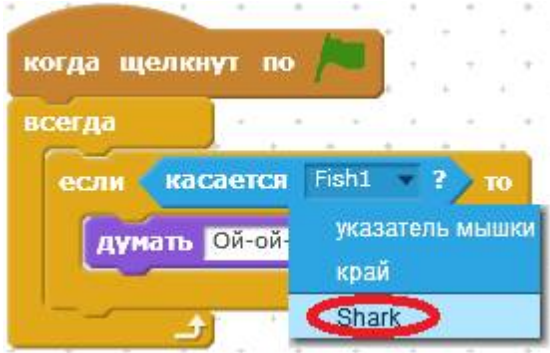
Теперь надо немного изменить разговорный блок, ведь рыбка не может съесть Акулу. Напишите там: «Ой-ой-ой!»



Запустите программу. Оба персонажа плавают вправо-влево, только Рыбка получилась очень уж трусливая. Она постоянно думает «Ой-ой-ой!», даже тогда, когда не касается Акулы. Наверное, в разговорном скрипте Рыбки есть какая-то ошибка.



Прочитайте второй скрипт внимательно. «Всегда, если Рыбка касается Fish1, то она думает про «Ой-ой-ой!» А что такое Fish1? Это же спрайт Рыбки! То есть Рыбка думает свою мысль, когда она касается сама себя, а это происходит постоянно! Надо изменить голубой блок. Нажмите на маленький чёрный треугольник и выберите там Shark — спрайт Акулы.



Вот так должна выглядеть правильная программа для Рыбки.



Запустите программу. Теперь всё работает так, как было задумано.

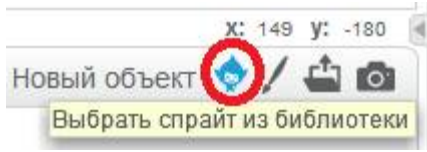


Маленький мультик готов. Не забудьте сохранить его.

Задание для самостоятельной работы. Добавьте в мультик ещё одну рыбку, которая будет плавать и о чем-нибудь мечтать.

ГЛАВА 5. ФУТБОЛ.

Давайте научим Котёнка играть в мячик. Создайте новый пустой проект, и добавьте из библиотеки спрайтов Мяч. Выберите его и нажмите Ок.



Теперь у нас два спрайта — Котик и Мяч.

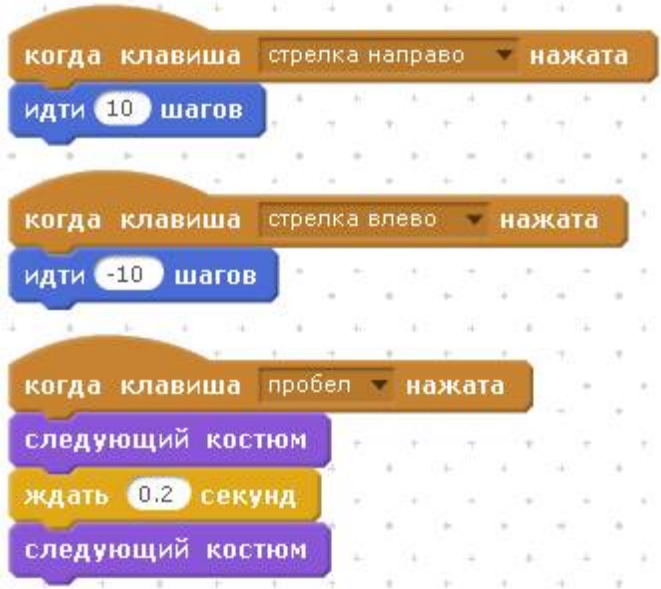


Выберите Котёнка.

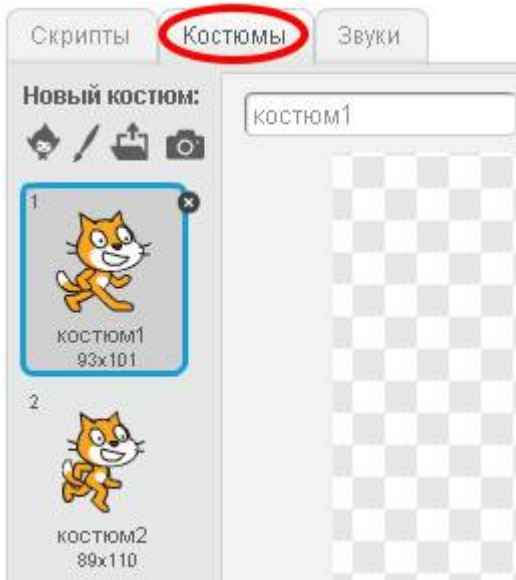


Сделайте ему такую программу из трёх скриптов.





Здесь использован новый фиолетовый блок «следующий костюм». Нажмите на вкладку «Костюмы», и вы увидите, какие костюмы есть у Кота.

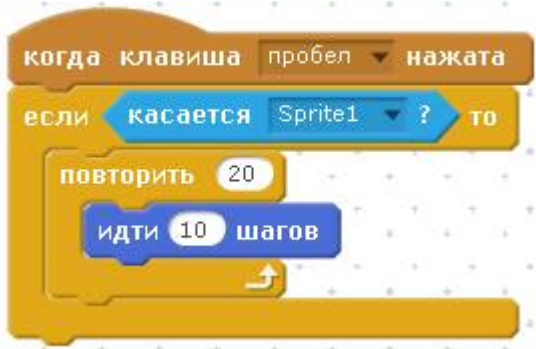


Сейчас у Котёнка два разных костюма. Если быстро поменять эти костюмы, то будет казаться, что Котик ударил ногой по Мячу. Для смены костюма используется блок «следующий костюм». Между сменами костюма пройдёт 0.2 секунды — одна пятая часть секунды. Подробнее про числа с точкой рассказано в приложении 5. Движением Кота вправо и влево управляем с помощью соответствующих стрелок. Если нажимаем стрелку направо — он идёт 10 шагов направо. Если нажимаем стрелку влево — он идёт

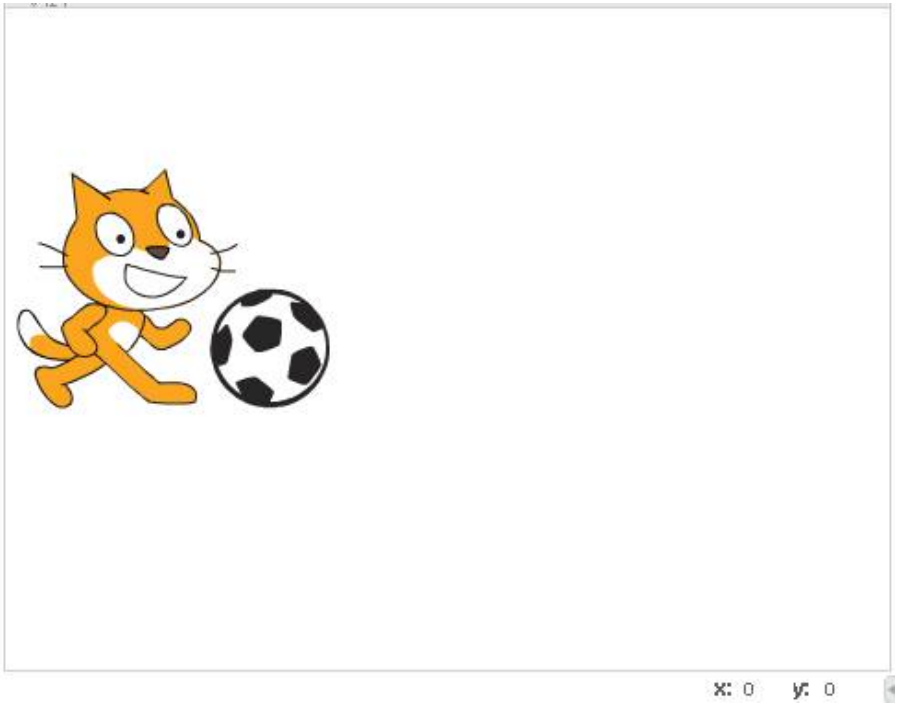
10 шагов налево. Для изменения направления движения использовано число со знаком минус. Минус означает, что это отрицательное число — десять наоборот. Поэтому Котик идёт не направо, а наоборот — налево. Подробнее про отрицательные числа рассказывается в приложении 3. Побегайте Котёнком, понажимайте пробел. Мячик никак не реагирует на удары. Надо запрограммировать его! Выберите Мяч.



Сделайте ему такой скрипт.



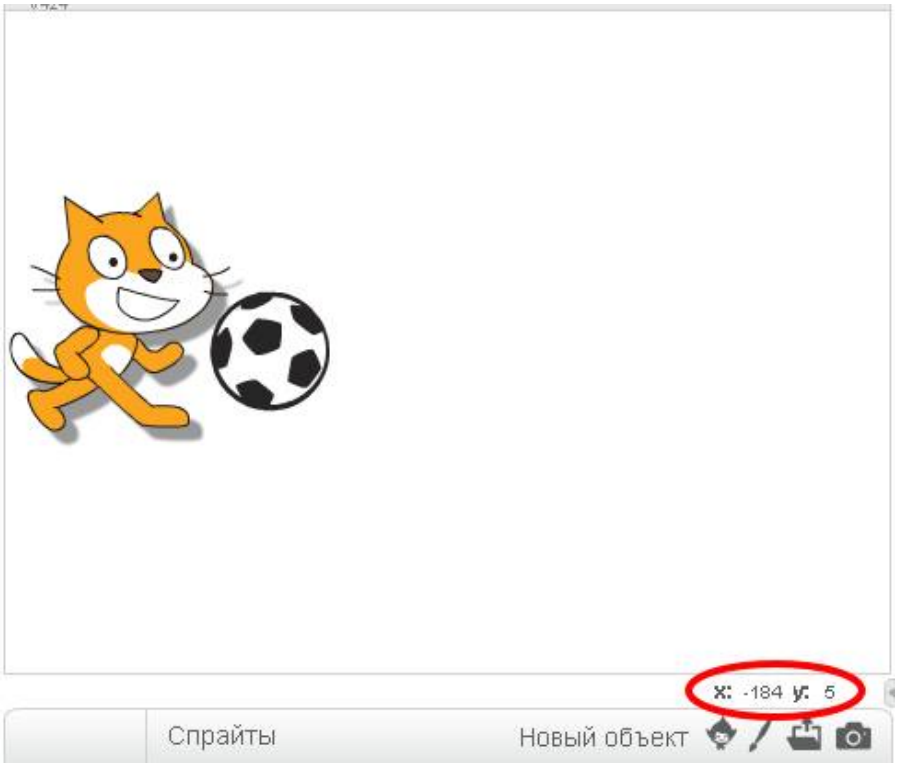
Подвиньте Мячик к Котенку так, чтобы они соприкасались. Нажмите на пробел. Мячик пролетел 20 раз по 10 шагов (всего 200 шагов) и остановился. Подойдите к нему Котиком и снова нажмите пробел. Снова удар и Мяч летит дальше до самой границы экрана. Нажимайте на стрелку влево, Котёнок вернётся в левую часть экрана. Мышкой перетащите к нему Мячик. Снова нажмите пробел — Мяч опять полетел. Получилась очень простая игра. Давайте её улучшим. Сделаем так, чтобы при нажатии на стрелку вверх Мяч и Котёнок автоматически возвращались на исходную позицию. Для этого давайте познакомимся с координатами Икс и Игрек (X и Y). Посмотрите внимательно в правый нижний угол сцены — там видны буквы X и Y, а также какие-то числа, которые изменяются при движении мышки. Эти числа — координаты точек сцены. У каждой точки сцены есть точные координаты.



Попробуйте очень аккуратно переместить курсор в центр сцены.



Координаты центральной точки сцены X = 0 и Y = 0. Теперь кликните на Котёнка и перемещайте его в левую часть экрана. Запомните его координаты.



Теперь запомните координаты Мяча.



Добавьте Котёнку скрипт перемещения на исходную позицию. В программе у Кота теперь четыре скрипта.

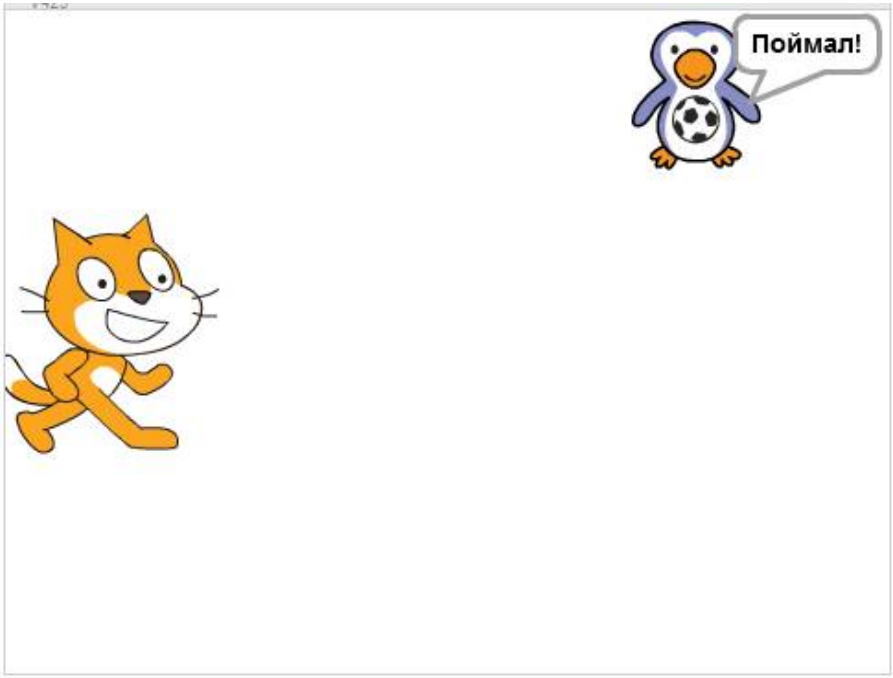


В программу к Мячу добавьте скрипт с его точными координатами. Программа Мяча станет такой.

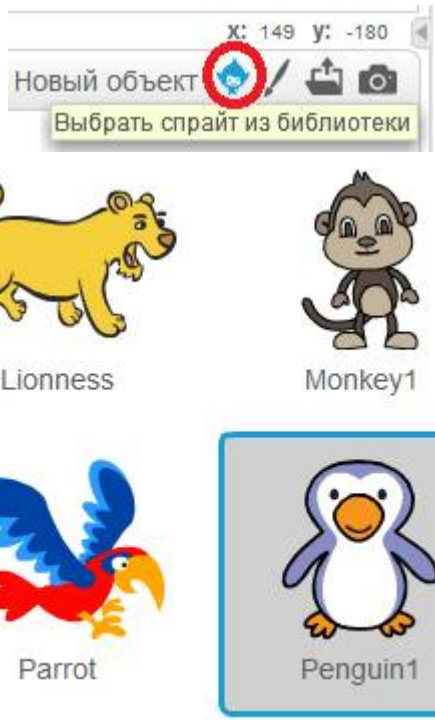

когда клавиша пробел нажата
если касается Sprite1 ? то
повторить 20
идти 10 шагов
когда клавиша стрелка вверх нажата
перейти в x: -125 y: -30

Теперь играть стало проще, когда Мячик снова улетит на край экрана, просто нажмите на стрелочку вверх, спрайты Кот и Мяч переместятся в исходное состояние.

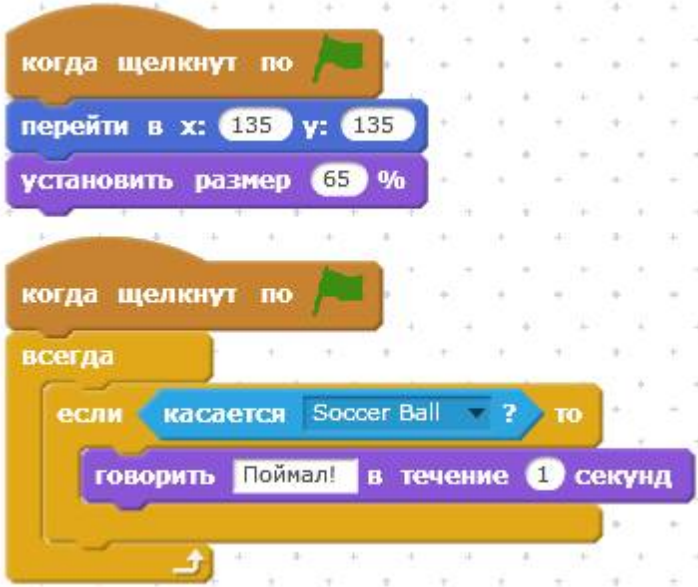
Давайте сделаем ещё несколько доработок. Для того чтобы игра стала похожа на настоящий футбол, надо добавить вратаря.



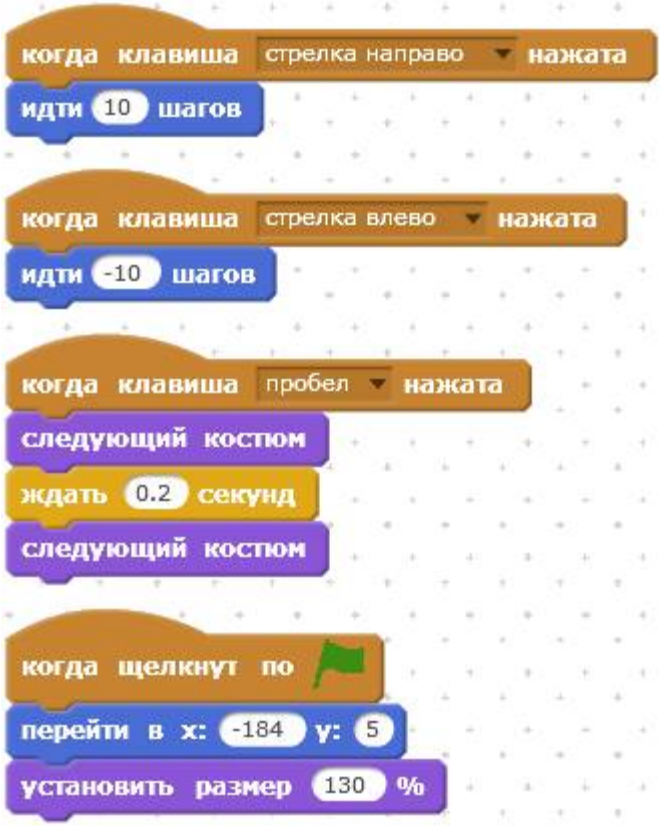
Выберите кого-нибудь из библиотеки спрайтов, например, Пингвина, и нажмите Ок.



Переместите Пингвина в правый верхний угол сцены и сделайте ему такую программу.


когда щелкнут по
перейти в x: 135 y: 135
установить размер 65 %
когда щелкнут по
всегда
если касается Soccer Ball ? то
говорить Поймал! в течение 1 секунд

Здесь использован новый фиолетовый блок «установить размер 65%». Этот блок уменьшает размер Пингвина, как будто он стоит вдалеке. Подробнее о процентах можно узнать в приложении 7. Программы Котёнка и Мяча тоже нуждаются в доработке. Добавьте в программу Котёнка блок «установить размер 130%». Этот блок немного увеличит его размер, как будто Котик стоит недалеко от нас. Стрелку вверх больше не используем. Программа будет запускаться зелёным флажком.



Программа Мяча тоже требует доработки. Теперь при ударе по Мячу он будет поворачиваться в направлении 60 градусов, и лететь, постоянно уменьшаясь в размере. Будет похоже на то, как Мяч удаляется от нас и летит к стоящему вдалеке вратарю. Когда Мяч долетит до вратаря, его размер станет совсем маленьким. При повторном запуске программы с помощью зелёного флажка размер Мяча снова станет нормальным — он будет установлен в 100%.



Программа готова. Можете поиграть в футбол.

Иногда, долетая до вратаря, Мяч прячется за него. Это не правильно. Выберите Мяч и два раза кликните на фиолетовом блоке «перейти в верхний слой». Теперь Мяч всегда будет сверху.

Вопрос. Как ускорить полёт мяча?



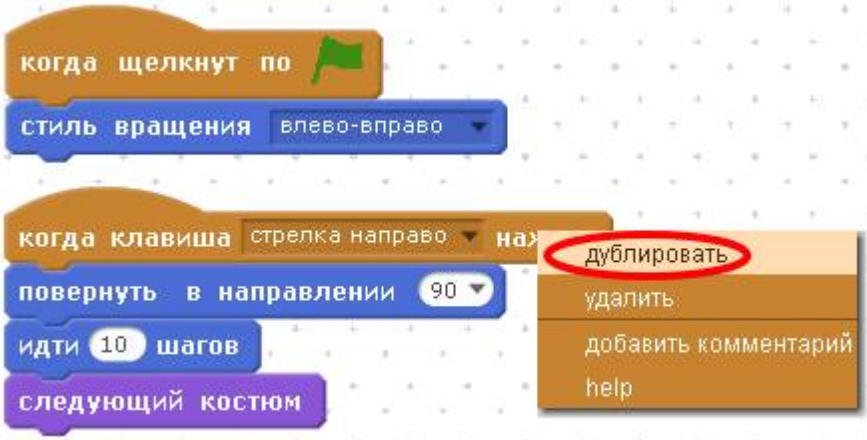
Задание для самостоятельной работы. В программе Мяча переместите блок «повернуть в направление 60 градусов» в скрипт к зелёному флажку, и добавьте ещё два скрипта для управления направлением Мяча с помощью стрелочек вверх и вниз. Теперь он не всегда будет лететь к Пингвину, и у Кота появится шанс забить гол!

ГЛАВА 6. ЗНАКОМИМСЯ С КООРДИНАТОЙ X.

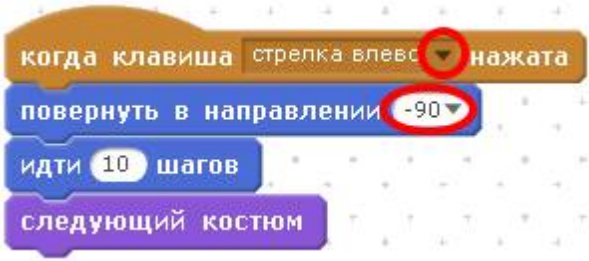
Создайте новый проект. Сделайте Коту такую программу. При нажатии на стрелку вправо Кот будет идти направо, забавно перебирая ножками. Как работают все эти блоки вы уже знаете.



Теперь запрограммируйте стрелку влево. Самый простой способ сделать это — дублировать скрипт стрелки вправо и немного изменить числа. Кликните правой кнопкой на блок «когда клавиша стрелка направо нажата» и выберите команду «дублировать».



Внесите небольшие изменения в скрипт. Измените стрелку и направление движения.



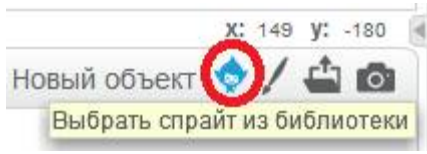
Теперь Кот управляется стрелками и может ходить направо и налево. Если при движении налево Кот перевернулся вверх ногами, значит, Вы забыли нажать на зелёный флажок, который в нашей программе задаёт стиль вращения. А сейчас давайте добавим в проект Гигу и запрограммируем её движение с использованием координаты X. Найдите Гигу в библиотеке.



Шутка.



Библиотека на своём месте, и, судя по рисунку, Гига работает библиотекарем!



Посмотрите, как улыбается Гига, она рада, что скоро сможет погулять.



Fish3



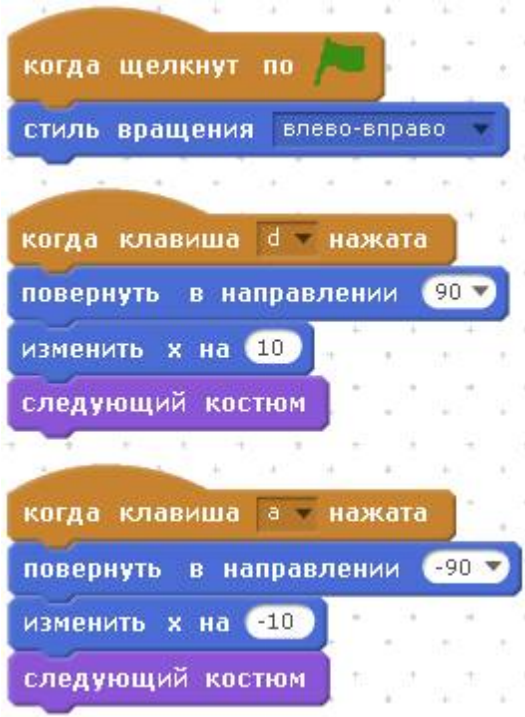
Football





Girl1

Сделайте ей такую программу.



Гига управляется клавишами a и d на клавиатуре.

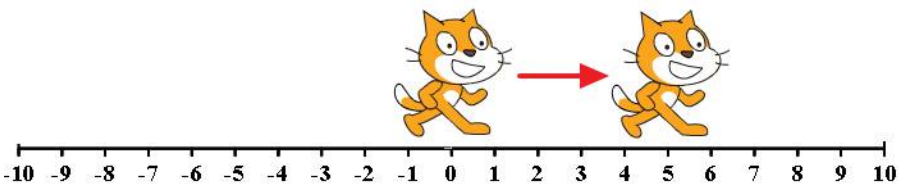


Не забудьте переключить язык на английский, а то программа работать не будет!

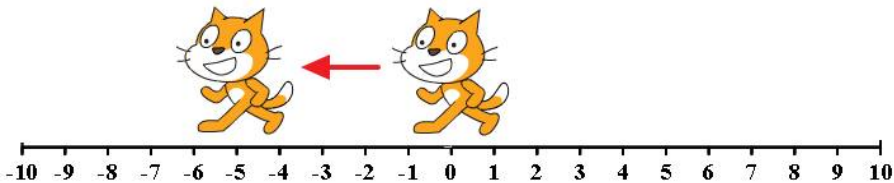
Вместо блоков «идти 10 шагов» мы использовали блоки «изменить X на 10». Следовательно, изменяя координату X любого спрайта мы можем заставить его двигаться вправо или влево. Если X изменяется на положительное число, спрайт двигается вправо. Если X изменяется на отрицательное число, то спрайт перемещается влево. Графически это можно нарисовать в виде горизонтальной линии с числами — в виде оси X.



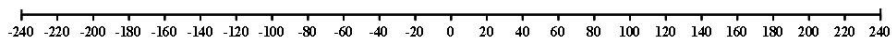
Например, если в начале программы Кот перемещён в точку с координатой X равной нулю, а затем получил команду «изменить X на 5», то он переместиться в точку с координатой X равной пяти, пойдёт вправо.



И наоборот, если в начале программы Кот перемещён в точку с координатой X равной нулю, а затем получил команду «изменить X на -5», то он переместиться в точку с координатой X равной минус пяти, пойдёт влево.



Координата X в Scratch может изменяться от -240 до 240.



Если дать спрайту задание перейти в точку с координатой X больше 240, то спрайт скроется за краем экрана, а если координата X станет больше миллиарда, то спрайт исчезнет в неизвестном направлении.



Вопрос: «Сколько нулей у миллиарда?»

Давайте сделаем так, чтобы Гига не исчезала за краем сцены, а появлялась с другой стороны и продолжала шагать, как ни в чём не бывало.

Добавьте Гиге в программу следующий интересный скрипт. В нём использованы овальные блоки положения по оси X и зелёный оператор сравнения. Как же работает этот скрипт? А очень просто. Если Кот дойдёт до края экрана, то его координата X станет равна 240. Если он сделает еще один маленький шажок, и координата X станет больше 240, то сработает блок «если» и значение переменной X будет установлено в -240. Кот телепортируется на противоположный край экрана и продолжит бежать направо. Подобная телепортация произойдёт и при пересечении левой границы экрана.



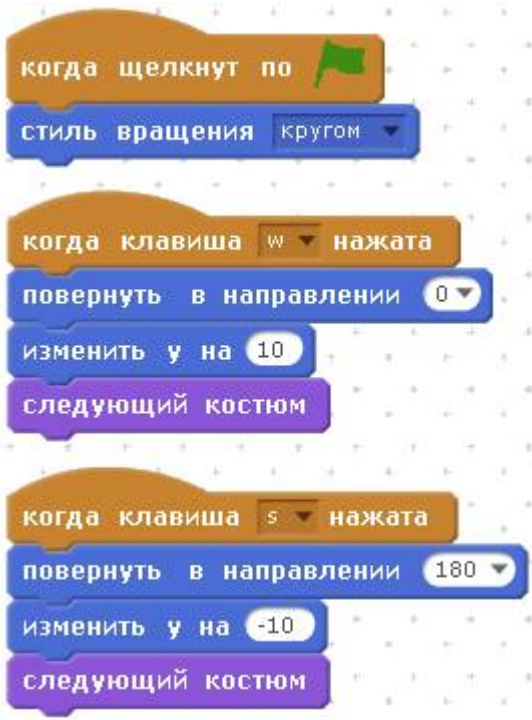
Надеюсь, вы поняли, как перемещать персонажей с помощью координаты X, давайте теперь познакомимся с координатой Y.

ГЛАВА 7. ЗНАКОМИМСЯ С КООРДИНАТОЙ Y.

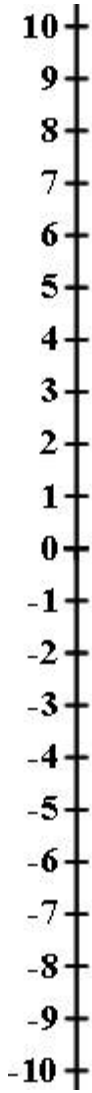
Создайте новый проект. Сделайте Коту такую программу. При нажатии на стрелку вверх Кот будет идти вертикально вверх, при нажатии стрелки вниз Кот будет идти вниз.



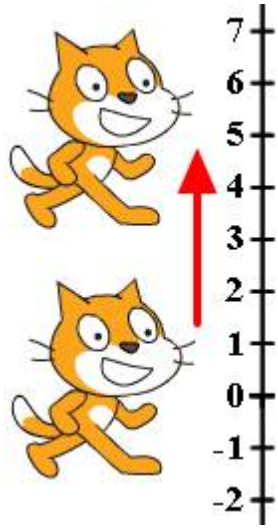
Программу для Гиги сделайте с применением блоков «изменить Y». Гига будет управляться клавишами w и s.



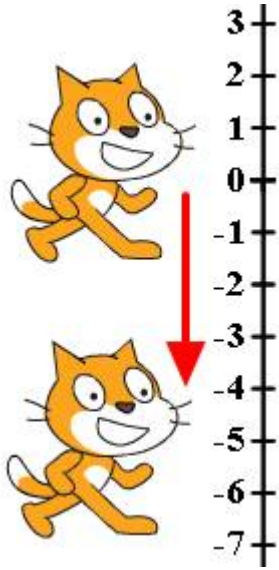
Как вы прекрасно видите, изменяя координату Y любого спрайта, мы можем заставить его двигаться вверх или вниз. Если Y изменяется на положительное число, спрайт двигается вверх. Если Y изменяется на отрицательное число, то спрайт перемещается вниз. Графически это можно нарисовать в виде вертикальной линии с числами — в виде оси Y.



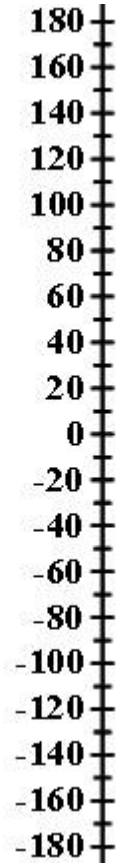
Например, если в начале программы Кот перемещён в точку с координатой Y равной нулю, а затем получил команду «изменить Y на 5», то он переместиться в точку с координатой Y равной пяти, поднимется вверх.



И наоборот, если в начале программы Кот перемещён в точку с координатой Y равной нулю, а затем получил команду «изменить Y на -5», то он переместиться в точку с координатой Y равной минус пяти, опуститься вниз.



Координата Y в Scratch может изменяться от -180 до 180.



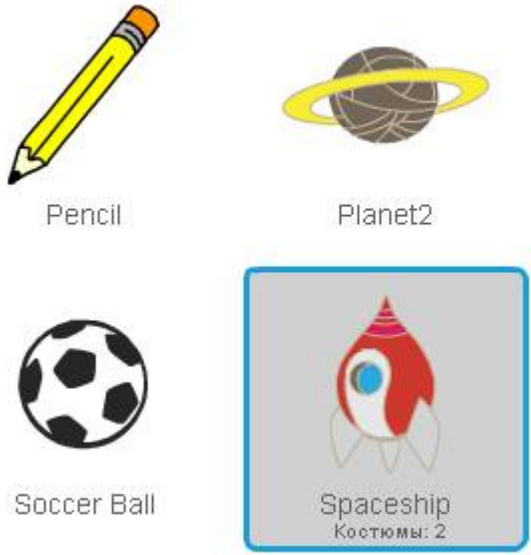
Если дать спрайту задание перейти в точку с координатой Y больше 180, или меньше -180, то спрайт, как вы, наверное, догадались, скроется за краем экрана.

Давайте снова сделаем так, чтобы Гига не исчезала за краем сцены, а появлялась с другой стороны и продолжала шагать, как ни в чём не бывало.

Добавьте Гиге в программу следующий скрипт. В нём использованы овальные блоки положения по оси Y и зелёный оператор сравнения. Как работает этот скрипт, вы уже знаете.



А теперь давайте добавим Ракету, которая будет взлетать при нажатии на пробел. Выберите её из библиотеки.



Сделайте Ракете такой скрипт.



Нажмите на пробел, Ракета полетит вверх и продвинется 25 раз по 10 пикселей вверх и исчезнет. Пиксели — это то же самое, что и точки экрана.

Вопрос. Как ускорить полёт ракеты?



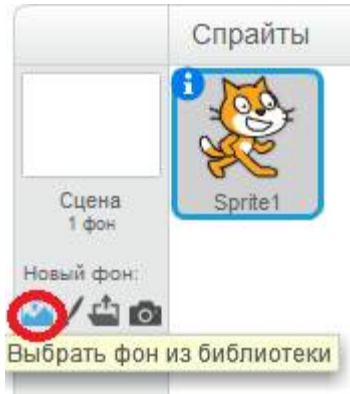
Задание. Добавьте любой спрайт и сделайте ему такую программу, чтобы при нажатии на зелёный флажок он появлялся вверху экрана и плавно падал вниз.

ГЛАВА 8. МУЛЬТИК «ЛЕТУЧИЙ КОТ И ЛЕТУЧАЯ МЫШЬ».

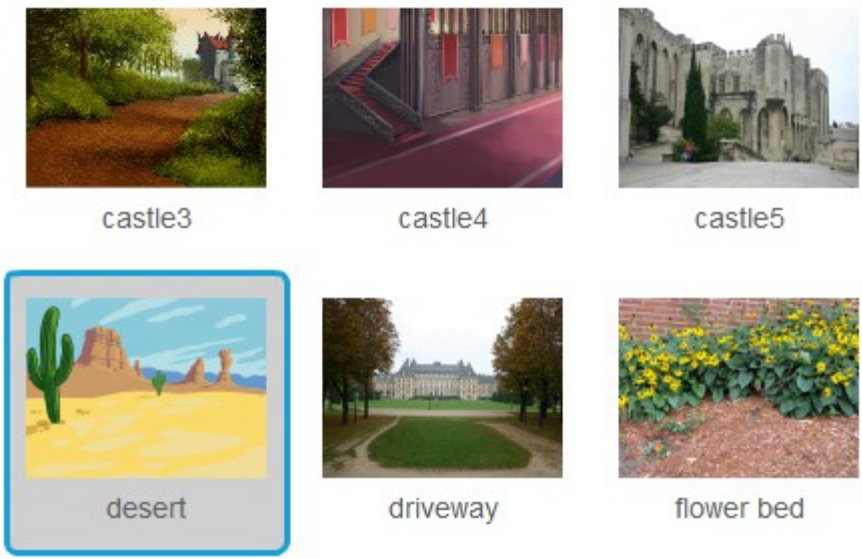
Давайте сделаем маленький мультик про Кота, который гулял по пустыне, повстречался с Летучей мышью, и улетел от неё на реактивном ранце. Летучая мышь заплачет горькими слезами. В этом проекте мы узнаем о том, как загрузить фон на стену, что такое костюмы персонажей, познакомимся с координатами X и Y, а также научимся рисовать в графическом редакторе. Готовый проект будет выглядеть примерно так.



Создайте новый проект. Фон нового проекта всегда белый. Первым делом надо добавить красивый фон. Фон будет располагаться на сцене. Нажмите на кнопку «Выбрать фон из библиотеки».



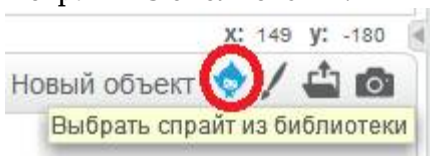
Откроется библиотека фонов. Выберите фон «desert» и нажмите ОК.



Теперь проект выглядит вот так.



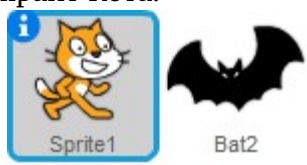
Добавьте второго персонажа — летучую мышь. Нажмите на кнопочку «Выбрать спрайт из библиотеки».



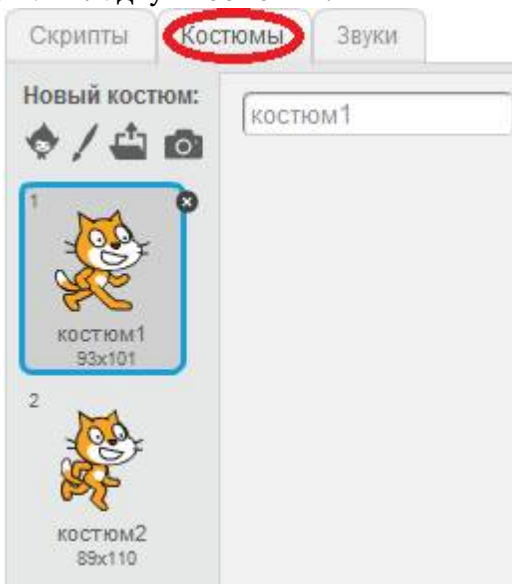
Выберите Летучую мышь и нажмите ОК. Растащите персонажей. Кота влево, Летучую мышь вправо.



Теперь надо добавить нашим персонажам по несколько костюмов. Выберите спрайт Кота.



Перейдите на вкладку «Костюмы».

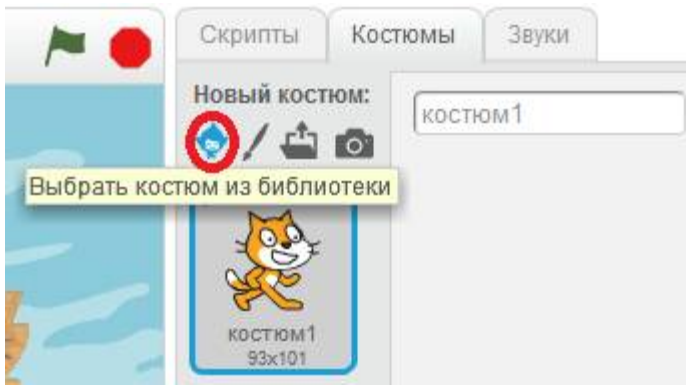


Откроется окно графического редактора. Здесь можно

рисовать, изменять существующие костюмы и рисовать новые. Как Вы видите, сейчас у Кота, как обычно, два костюма. В этом мультфильме второй костюм нам не нужен. Удалите его крестиком. Останется только костюм1.

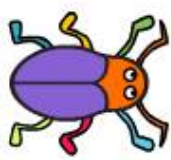


Теперь добавьте Коту ещё один костюм, такой же, как и костюм1.



Выберите Кота и нажмите ОК.

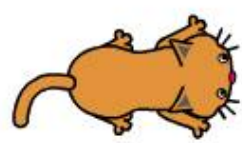

bear2


beetle

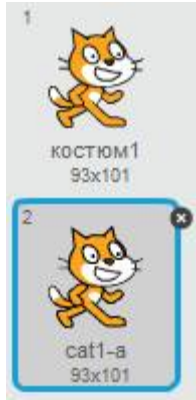

butterfly1-a




cat1-b


cat2

Теперь у Кота два одинаковых костюма.



Подрисуем Коту на втором костюме реактивный ранец. Сначала, для того, чтобы было удобнее рисовать, надо увеличить Кота. Два раза нажмите на кнопку плюс.



Кот станет огромным.



Теперь надо нарисовать Коту реактивный ранец. Нажмите на кнопку рисования прямоугольника.



Выберите зелёный цвет.



Выберите закрашенный прямоугольник.



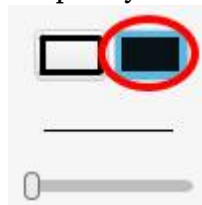
Нарисуйте Коту на спине реактивный ранец — зелёный прямоугольник.

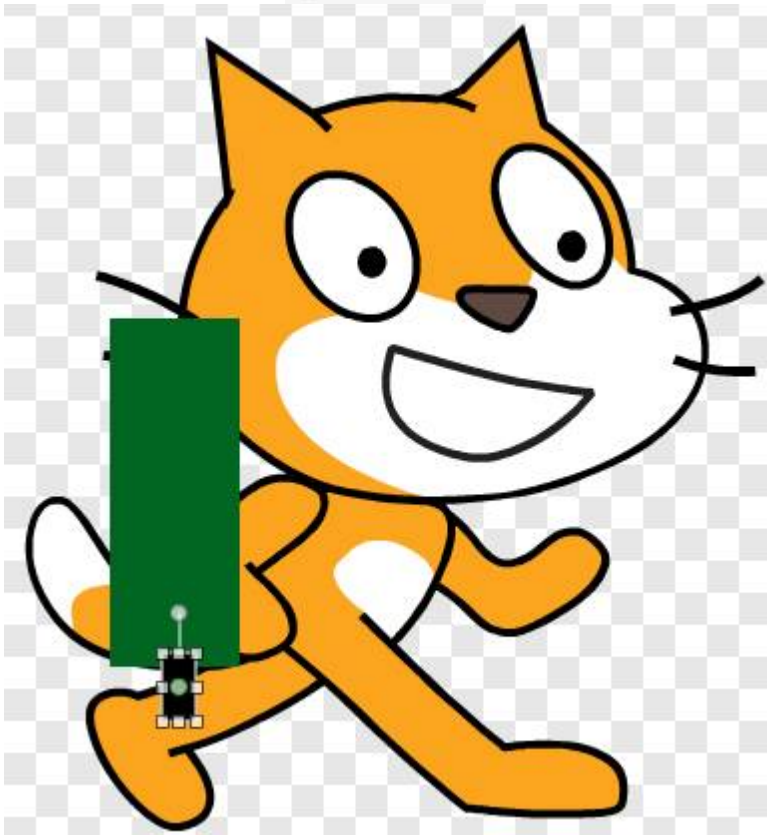


Потом выберите пипеткой чёрный цвет и подрисуйте сопло.

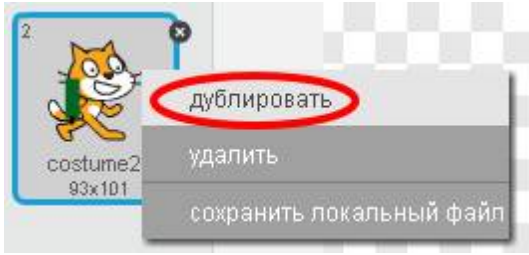


Выберите закрашенный прямоугольник.



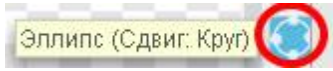


Теперь скопируйте второй костюм. Нажмите на него правой кнопкой мышки и выберите «дублировать».



Появится третий костюм, точная копия второго. Кликните на третий костюм и отредактируйте его. Подрисуйте пламя реактивного ранца.

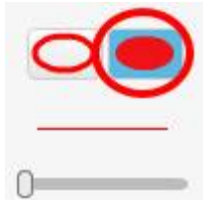
Выберите инструмент «Эллипс». Эллипс в переводе на русский означает что это овал. Фраза в скобках — (Сдвиг: Круг) означает, что при выборе этого инструмента и одновременном нажатии клавиши Shift на клавиатуре будет рисоваться не овал, а ровный круг.



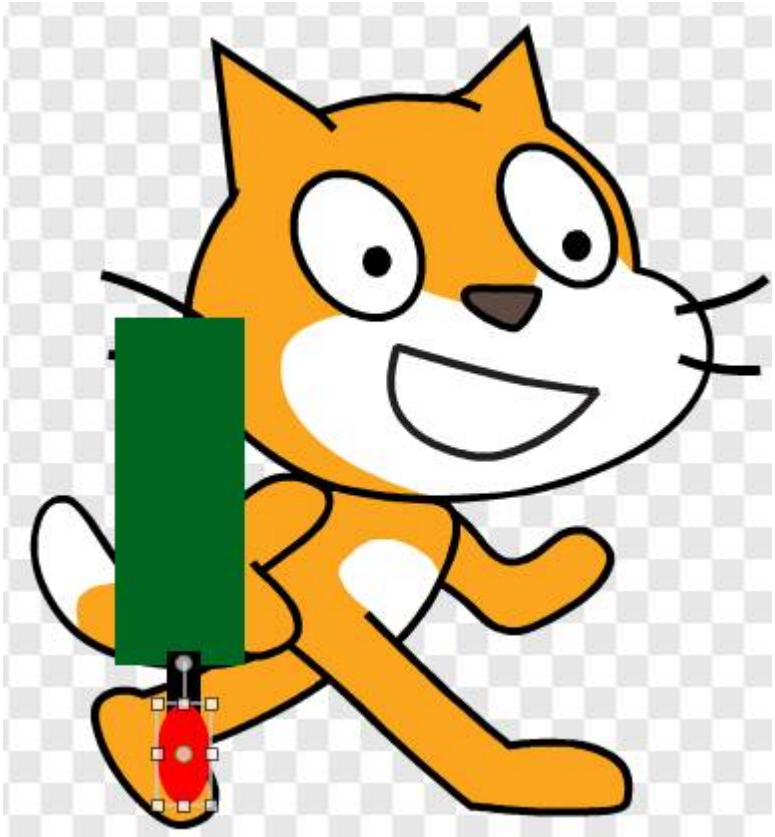
И выберите пипеткой красный цвет.



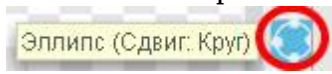
Выберите закрашенный прямоугольник.



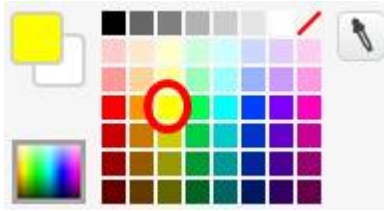
Подрисуйте огонь в виде красного овала.



В центре огня добавьте желтенького — нарисуйте там маленький жёлтый овал. Снова выберите инструмент «Эллипс».



Выберите жёлтый цвет. Закрашенный прямоугольник можно не выбирать, он и так выбран.



Нарисуйте жёлтый овальчик.

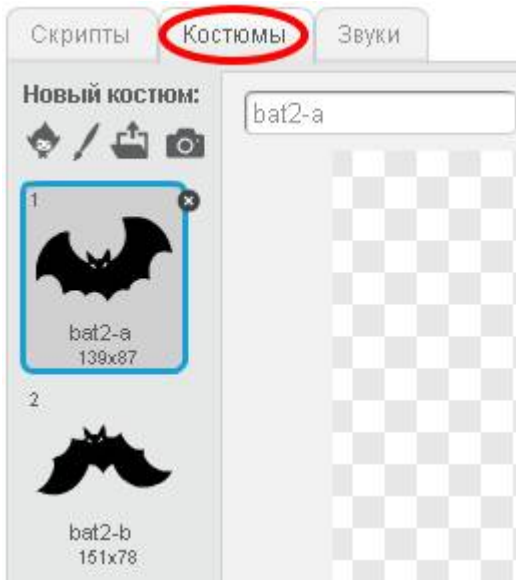


Огонь теперь как настоящий!

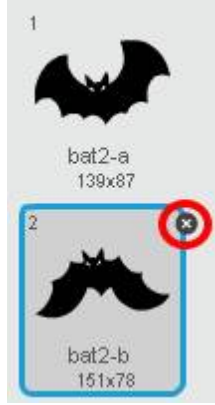
Летучей мыши тоже надо добавить один костюм. Выберите спрайт Летучей мыши.



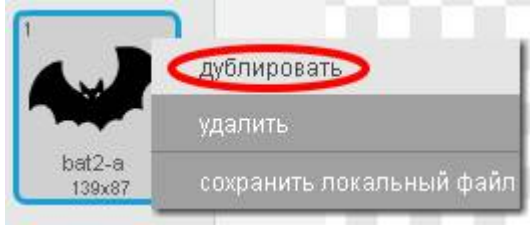
Перейдите на вкладку «Костюмы».



Удалите нижний костюм, нажав на маленький крестик.



Скопируйте первый костюм Летучей мыши, нажав на него правой кнопкой и выбрав «дублировать».



Отредактируйте новый костюм Летучей мыши. Сначала увеличьте масштаб - два раза нажмите на кнопку плюс.



Подрисуйте синенькие горькие слёзы. Выберите инструмент «Карандаш».



Выберите синий цвет.



Линия должна быть не самая тонкая.



Летучая мышь будет плакать и приговаривать: «Мой обед!»



Теперь у нас есть фон, Кот с тремя костюмами, и Летучая мышь с двумя костюмами. Сделайте программу Коту.



Давайте теперь внимательно посмотрим, как работает этот скрипт. Когда зелёный флажок будет нажат, Кот переместиться в левую часть сцены, сменит свой костюм на костюм без ранца и станет видимым. Затем Кот подождёт ровно одну секунду и плавно поплывёт ближе к летучей мыши, там он скажет: «Привет!» Затем подождёт три секунды, и скажет: «Пока!» Он сменит свой костюм на костюм с реактивным ранцем, подождёт одну секунду, и стартует, снова сменив костюм на костюм с включённым двигателем. Затем сорок раз Котик изменит свою координату Y на пять, то есть плавно поднимется вверх. В конце Котик исчезнет, Мышь будет огорчена…



Запустите программу зелёным флажком. Убедитесь что всё

работает правильно. Если Кот летает не так, как нужно, то проверьте значения координат X и Y, а также убедитесь что в программе использован блок «изменить Y на 5», а не «установить Y в 5», эти блоки расположены рядом, многие часто их путают.

Если Котик летает как надо, то сделайте программу летучей мыши.


когда щелкнут по
перейти в x: 136 y: -113
сменить костюм на bat2-a
ждать 5 секунд
говорить Я тебя съем! в течение 2 секунд
ждать 4 секунд
сменить костюм на bat2-a2
ждать 1 секунд
думать Мой обед!!!

Запускайте программу. Если все сделано правильно, Кот подойдет к летучей мыши, скажет: «Привет!». Мышь скажет: «Я тебя съем!». Кот наденет реактивный ранец, улетит вверх и исчезнет. Мышь заплачет: «Мой обед!».



Если что-то не работает, обратите внимание на названия костюмов и на числа в блоках. Некоторые числа с минусами. Это отрицательные числа. Подробнее об отрицательных числах рассказано в приложении 3.

Вопрос. Может ли Летучая мышь съесть кота? Чем питаются летучие мыши, живущие в северной части Африки.



Задание. Используя полученные знания сделайте маленький

мультик со своими персонажами.

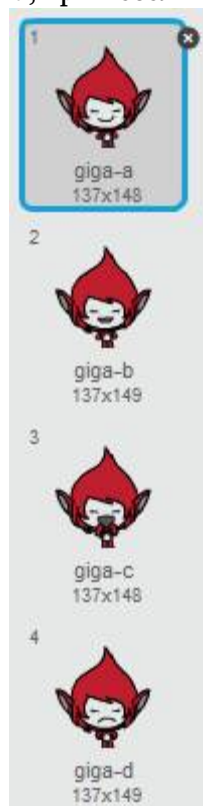
ГЛАВА 9. ИГРА ЛАБИРИНТ.

В этой главе мы сделаем игру в которой Гига должна будет пройти по лабиринту и найти Нано, который заблудился в пещере. Сначала мы сделаем простой вариант игры, а потом усложним его, добавив анимацию.

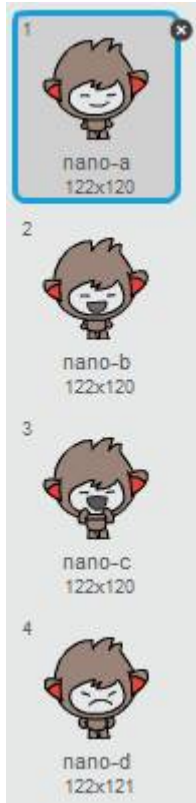
Создайте новый проект. Удалите Кота, и добавьте из библиотеки Гигу и Нано. Посмотрите, какие костюмы есть у Гиги. Выберите её и нажмите на вкладку Костюмы.



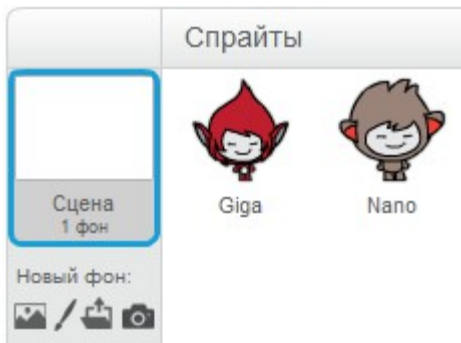
У Гиги четыре костюма, три весёлых и один грустный.



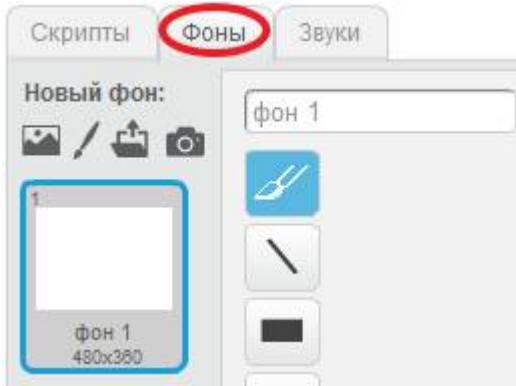
У Нано тоже четыре костюма, три весёлых и один грустный.



Запомните названия первых весёлых костюмов и названия грустных, они нам пригодятся. Затем нарисуйте лабиринт. Выберите сцену.



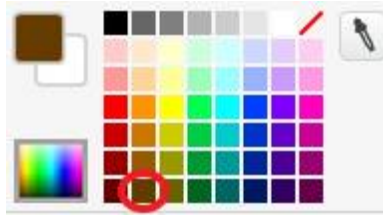
Перейдите на вкладку Фоны.



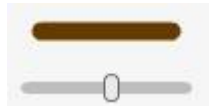
Выберите инструмент Линия.



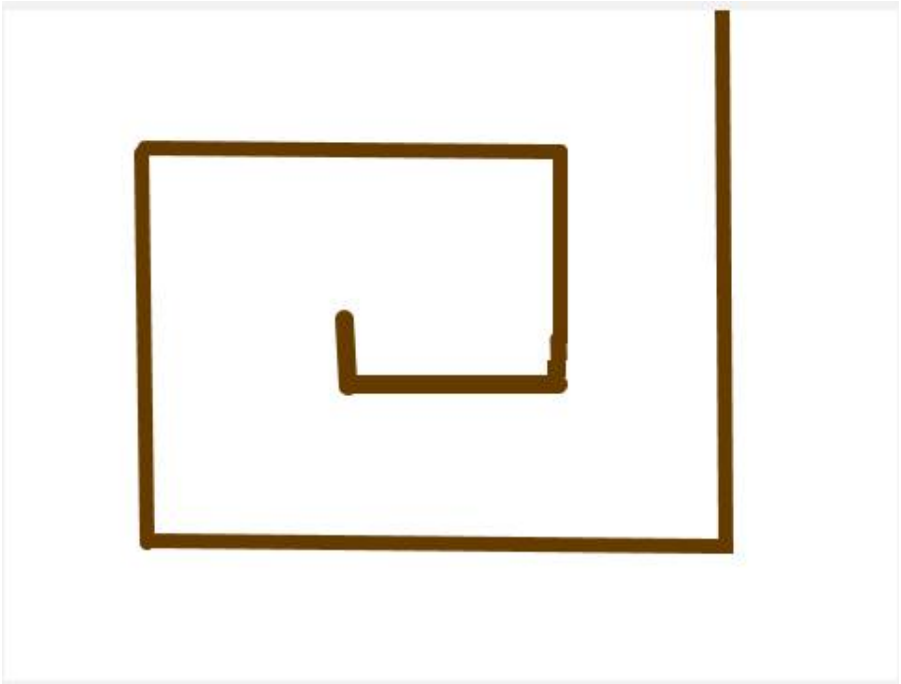
Выберите коричневый цвет.



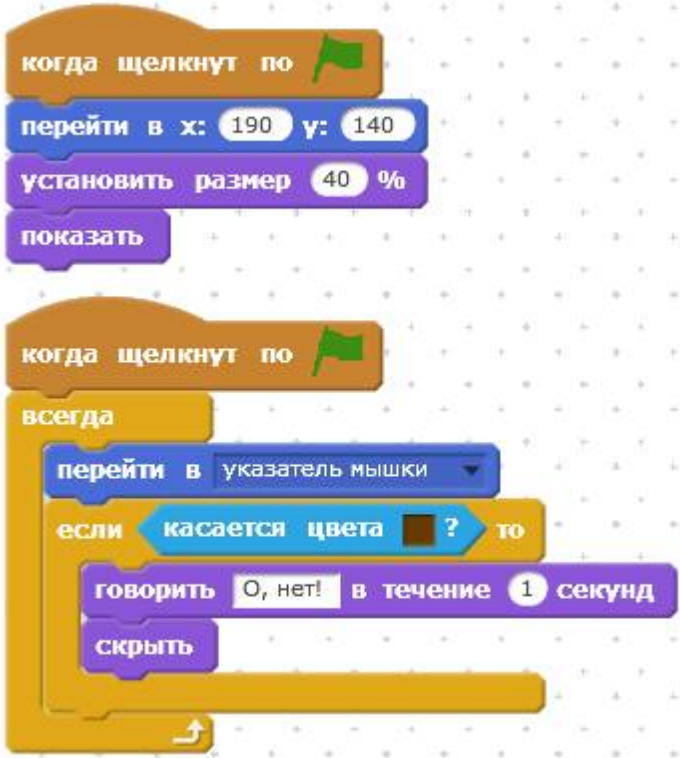
Увеличьте толщину линии.



Нарисуйте лабиринт.



Сделайте для Гиги такую программу. Здесь использованы знакомые вам блоки «перейти в указатель мышки», «показать», «скрыть», а также новый голубой сенсорный блок «касается цвета». Он используется совместно с блоком «если» и срабатывает в момент касания Гигой стенки лабиринта.



Для того чтобы поместить в голубой блок коричневый цвет используется очень простой способ. Кликните мышкой в квадратное окошко с цветом.



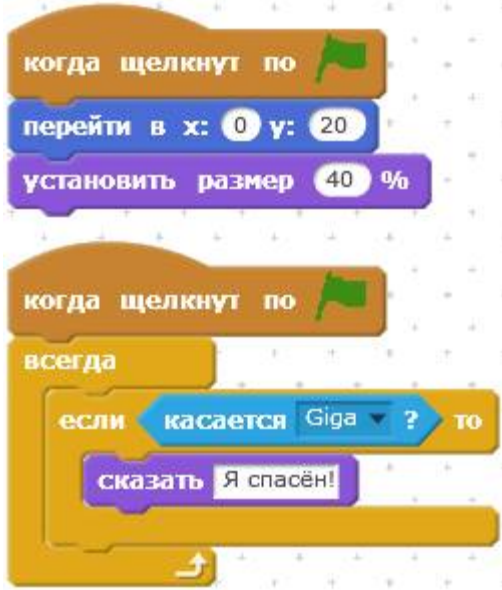
Курсор измениться, и станет похож на руку. Кликните им на стену лабиринта.



Выбранный цвет автоматически попадёт в голубой блок.



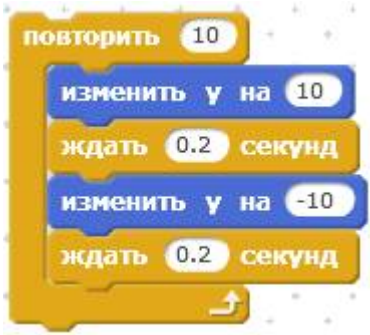
Теперь сделайте программу для Нано.



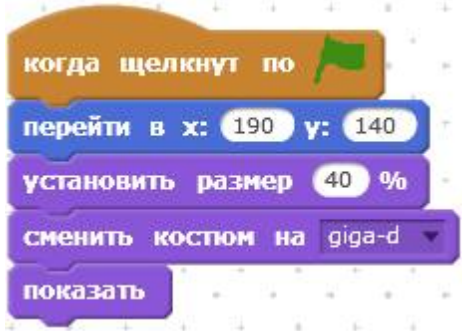
Запустите программу зелёным флажком, аккуратно двигайте Гигу по лабиринту. Старайтесь не касаться стен, а то проиграете. Когда вы дойдёте до Нано, то он обрадуется и скажет: «Я спасён!».

Теперь давайте добавим в нашу игру эмоций. Пусть в начале игры у Гиги и у Нано будут грустные выражения лиц, а в момент долгожданной встречи их озарит улыбка. А ещё пусть они запрыгают от радости.

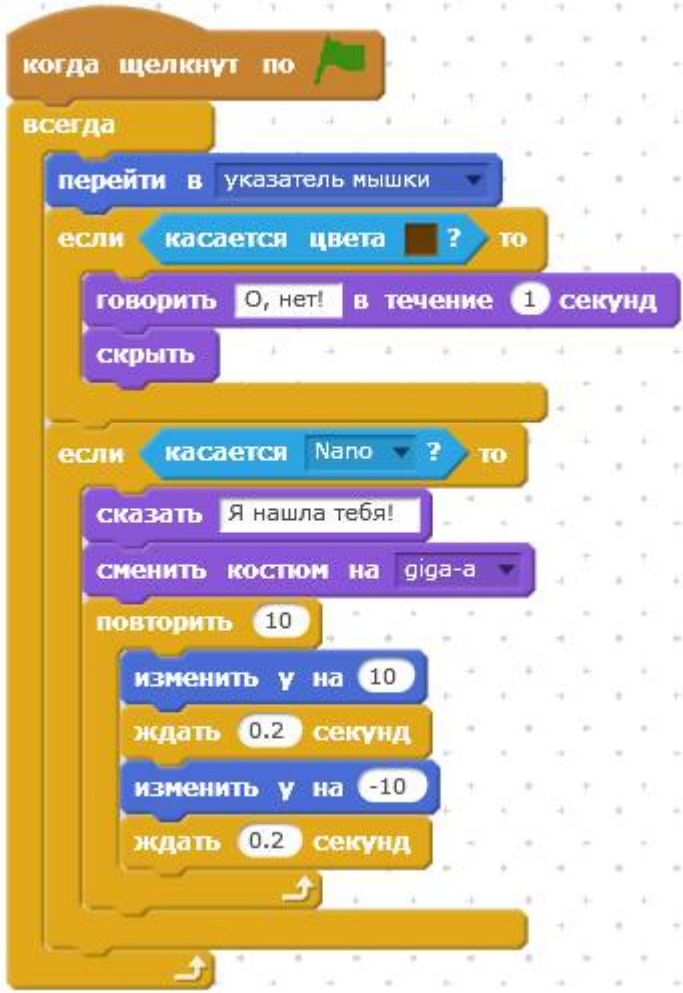
Для того, чтобы сделать Нано и Гигу грустными, надо добавить блок «сменить костюм на». В начале программы, когда нажат зелёный флажок, они должны сменить костюмы на грустные, а когда встретятся — на весёлые. Радостные прыжки персонажей можно сделать с помощью следующих блоков.



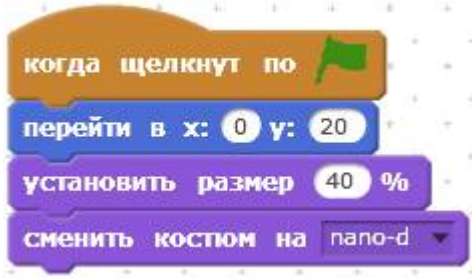
Этот кусочек скрипта работает следующим образом, при изменении Y на 10 персонаж подпрыгнет, подождёт одну пятую часть секунды и опустится вниз, изменив Y на -10. Опять подождёт 0.2 секунды, и снова подпрыгнет. Всего будет совершено 10 прыжков. В итоге первый скрипт программы Гиги станет такой.



А второй скрипт вот такой.



Первый скрипт Нано будет от такой.



А второй вот такой.



Проект готов. Нажмите на зелёный флажок и попробуйте пройти Гигой до центра лабиринта.

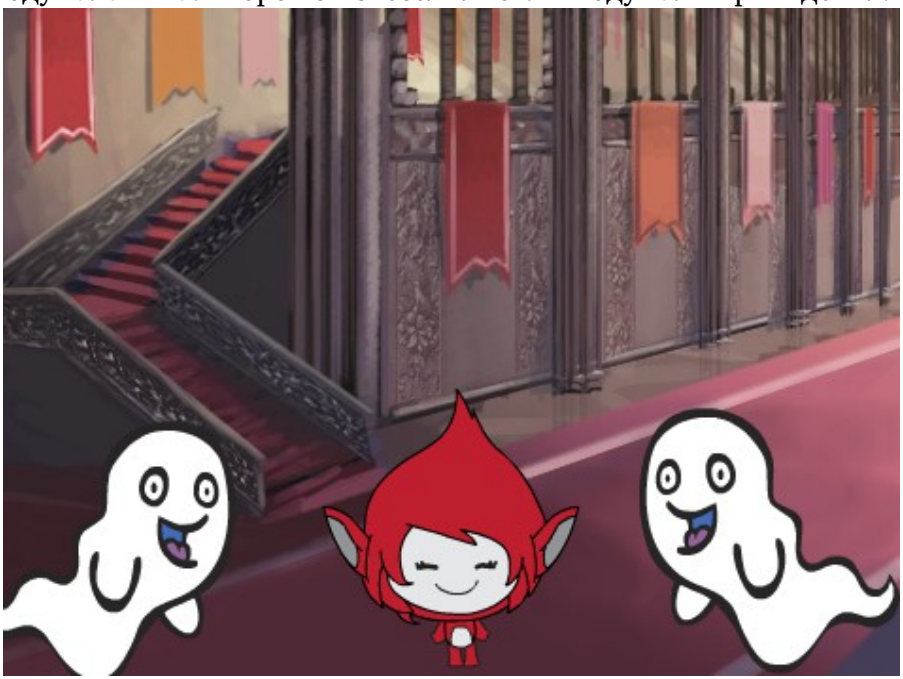
Вопрос. Что надо было сделать Нано, чтобы не заблудиться в пещере?



Задание. Усложните игру. Сделайте так, чтобы Гиге нельзя было касаться не только стен лабиринта, но и границ экрана.

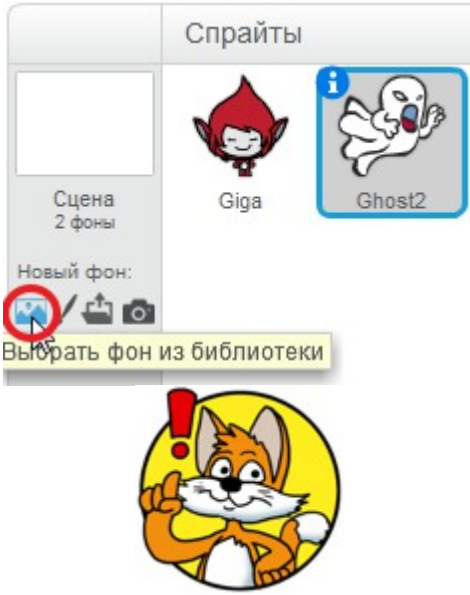
ГЛАВА 10. МУЛЬТИК С ПРИВИДЕНИЯМИ.

Как-то раз, ближе к вечеру, Гига отправилась побродить по коридорам старого замка. Вдруг откуда ни возьмись выплыли два привидения и испугали Гигу до полусмерти! Но через секунду привидения снова растворились в воздухе. «Показалось» - подумала Гига. «Хорошо повеселились!» - подумали привидения.

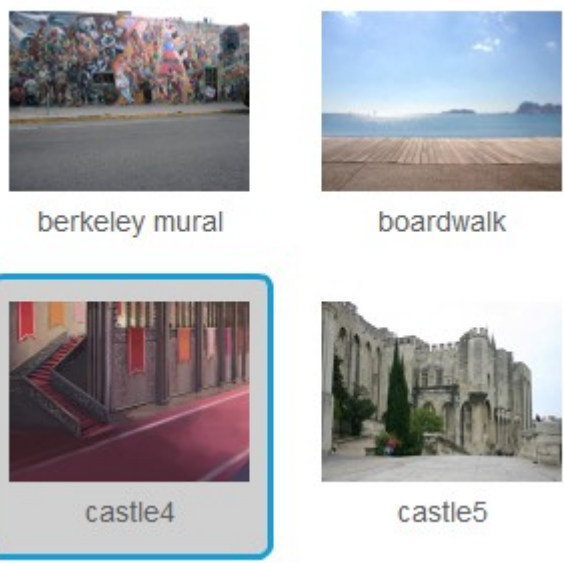


В этом проекте мы снова будем использовать координаты X и Y, различные эффекты, а ещё научимся дублировать спрайты и изменять внешний вид персонажей в графическом редакторе.

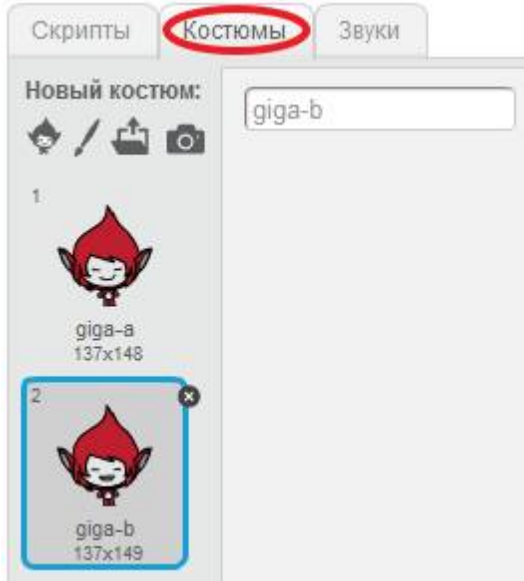
Создайте новый проект и удалите Кота. Добавьте из библиотеки спрайтов Гигу и одно Привидение. Затем выберите сцену, и выберите фон в виде коридора старого замка и нажмите Ок.



Обратите внимание, второе привидение добавлять не надо! Мы сделаем его потом продублировав первое привидение.



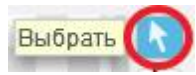
Разместите Гигу в центре нижней части экрана, а Привидение слева от неё. Теперь нам надо сделать испуганный костюм для Гиги. Выберите её и перейдите на вкладку «Костюмы». Выберите второй костюм.



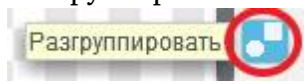
Для того чтобы было удобнее изменять костюм, его сначала надо увеличить. Нажмите два раза на плюсик.



Гига станет гигантской. Возьмите инструмент «Выбрать» и кликните в Гигу, она будет выбрана целиком, одновременно справа появятся несколько кнопок.



Кликните кнопку «Разгруппировать».



Гига моргнёт и немного разгруппируется. Снова выберите голову Гиги и нажмите кнопку «Разгруппировать». Теперь можно выбирать разные части Гиги по отдельности. Выберите инструмент «Изменение формы».



Кликните на левый глаз, он покроется точками привязки.



Потяните одну из нижних точек привязки чтобы увеличить глаз.

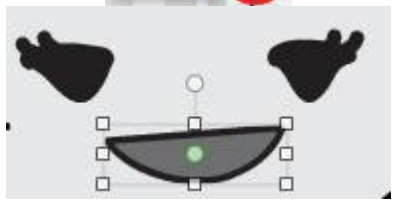


Аналогично увеличьте второй глаз.

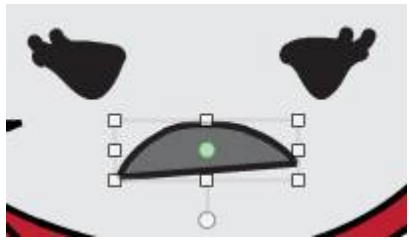


Теперь надо перевернуть рот. Кликните в него инструментом выбора.





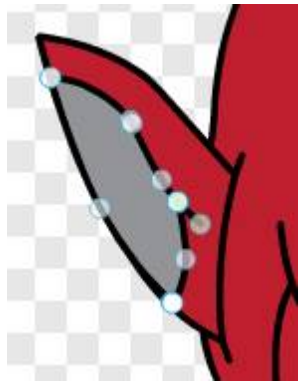
Затем возьмите мышкой за верхний кружочек и переверните рот.



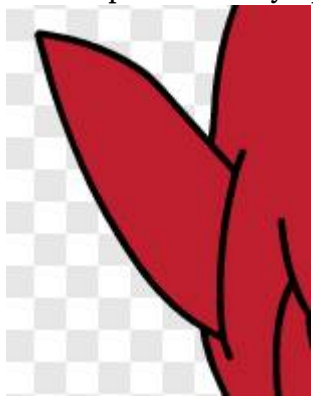
Какое грустное лицо! Давайте ещё изменим ушки. Кликните инструментом «Изменение формы» на серые вставки в ушах и удалите их кнопкой Del на клавиатуре.



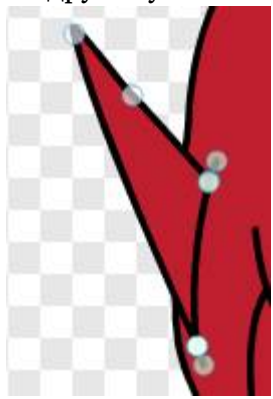
Станет так.



Теперь инструментом «Изменение формы» выберите само ухо и измените его переместив верхнюю точку привязки вправо.



Аналогично измените другое ухо.

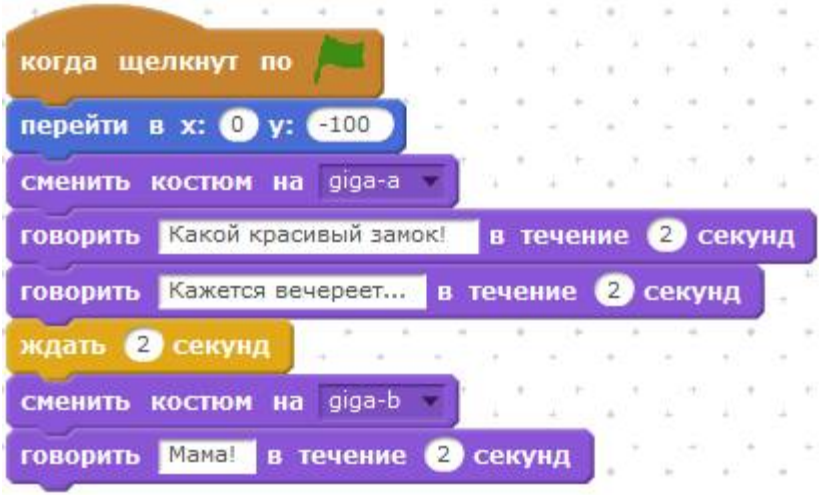




Теперь во втором костюме Гига выглядит достаточно испуганной.



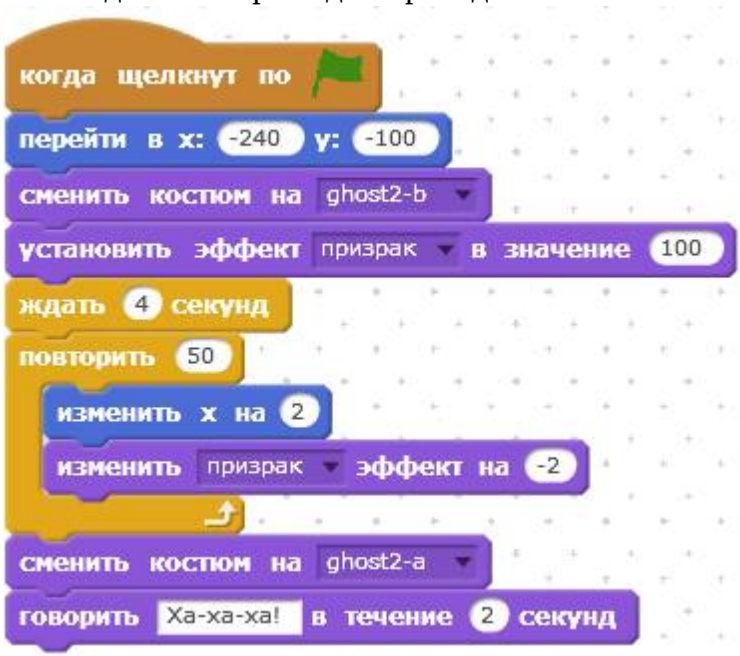
Можно начинать программировать поведение персонажей. Сделайте для Гиги такую программу.


когда щелкнут по
перейти в x: 0 y: -100
сменить костюм на giga-a
говорить Какой красивый замок! в течение 2 секунд
говорить Кажется вечереет... в течение 2 секунд
ждать 2 секунд
сменить костюм на giga-b
говорить Мама! в течение 2 секунд

Рассмотрим, как работает скрипт. Вначале Гига идёт в центр сцены и одевает свой нормальный костюм — мило улыбается. Она говорит первую фразу, затем вторую, а затем ждёт две секунды и меняет костюм на испуганный. В этот же момент она

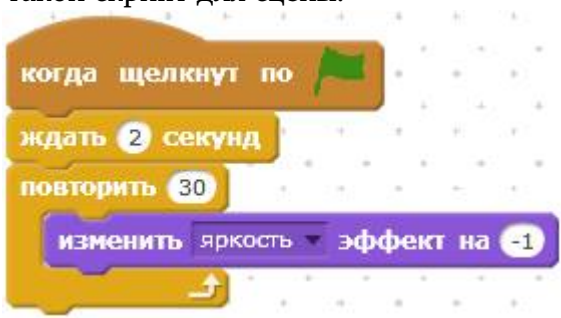
говорит: «Мама!»

Запустите программу. Убедитесь, что Гига всё делает правильно и сделайте скрипт для Привидения.


когда щелкнут по
перейти в x: -240 y: -100
сменить костюм на ghost2-b
установить эффект призрак в значение 100
ждать 4 секунд
повторить 50
изменить x на 2
изменить призрак эффект на -2
сменить костюм на ghost2-a
говорить Ха-ха-ха! в течение 2 секунд

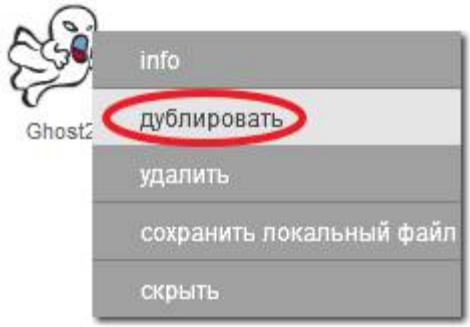
Посмотрим, как он работает. Сначала Привидение перейдёт в левую часть экрана, сменит костюм на милый, и станет полностью прозрачным из-за установки значения прозрачности в сто. Затем привидение ждёт четыре секунды и плавно плывёт к Гиге пятьдесят раз изменяя значение координаты X на два. Одновременно с движением происходит и плавное появление Привидения из ниоткуда, ведь значение эффекта «призрак» плавно уменьшается до нуля, до тех пор, пока Привидение не станет полностью прозрачным.

Запустите программу. Если всё работает так, как задумано, то сделайте вот такой скрипт для сцены.



Запустите программу зелёным флажком. Да уж, Гиге сейчас не позавидуешь. Для пущего страха давайте добавим второе Привидение справа. Самый простой способ сделать это —

дублировать первое Привидение и изменить его программу. Кликните на первое Привидение правой кнопкой мышки и выберите «дублировать».



Появится второе Привидение — точная копия первого и с такой же программой. Программу надо изменить. Давайте сделаем так, чтобы второе Привидение приближалось к Гиге справа. Для этого в скрипте надо изменить всего два числа. В итоге у второго Привидения должна получиться вот такая программа.

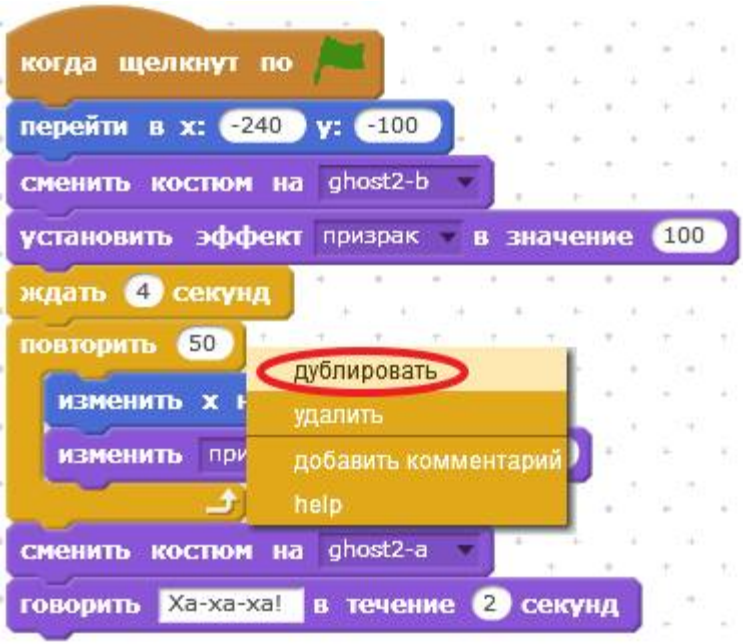

когда щелкнут по
перейти в x: 240 y: -100
сменить костюм на ghost2-b
установить эффект призрак в значение 100
ждать 4 секунд
повторить 50
изменить x на -2
изменить призрак эффект на -2
сменить костюм на ghost2-a
говорить Ха-ха-ха! в течение 2 секунд

Запустите программу. Второе Привидение теперь действительно приближается к Гиге справа, но как! Задом наперёд! Надо изменить направление второго Привидения. Добавьте в начало скрипта блок «повернуть в направление -90». Запустите программу. Что получилось? Привидение плывёт вверх ногами! Мы забыли установить стиль вращения! Добавьте в начало скрипта блок «стиль вращения влево-вправо». В итоге

должна получиться вот такая программа второго Привидения.


когда щелкнут по
стиль вращения влево-вправо
повернуть в направлении -90
перейти в x: 240 y: -100
сменить костюм на ghost2-b
установить эффект призрак в значение 100
ждать 4 секунд
повторить 50
изменить x на -2
изменить призрак эффект на -2
сменить костюм на ghost2-a
говорить Ха-ха-ха! в течение 2 секунд

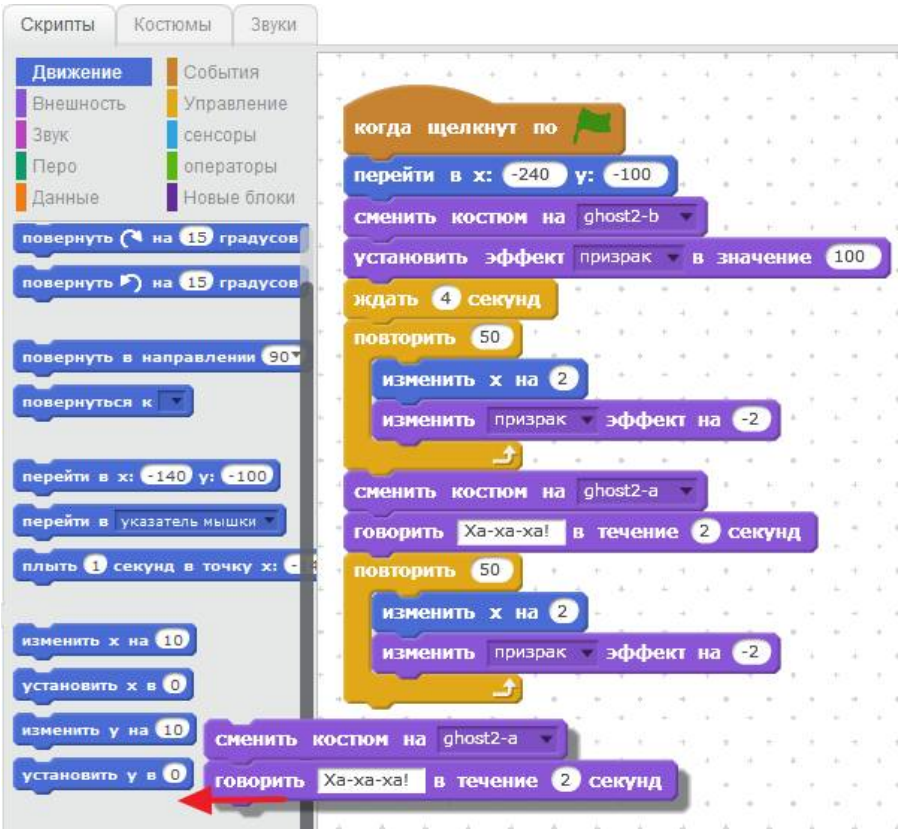
Запустите программу. Теперь всё нормально. У нас, но не у Гиги… Давайте спасём её от этого ужаса. Пускай привидения уплывут обратно и исчезнут. Выберите первое привидение и дублируйте блок «повторить 50» со всем содержимым.



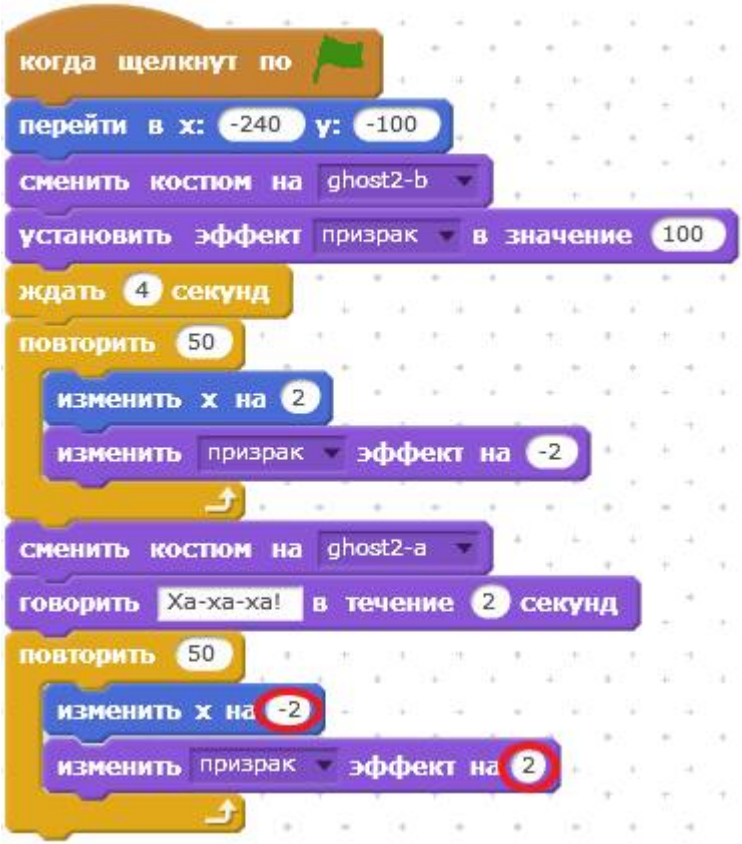
Прицепите скопированные блоки снизу к скрипту.


когда щелкнут по
перейти в x: -240 y: -100
сменить костюм на ghost2-b
установить эффект призрак в значение 100
ждать 4 секунд
повторить 50
изменить x на 2
изменить призрак эффект на -2
сменить костюм на ghost2-a
говорить Ха-ха-ха! в течение 2 секунд
повторить 50
изменить x на 2
изменить призрак эффект на -2
сменить костюм на ghost2-a
говорить Ха-ха-ха! в течение 2 секунд

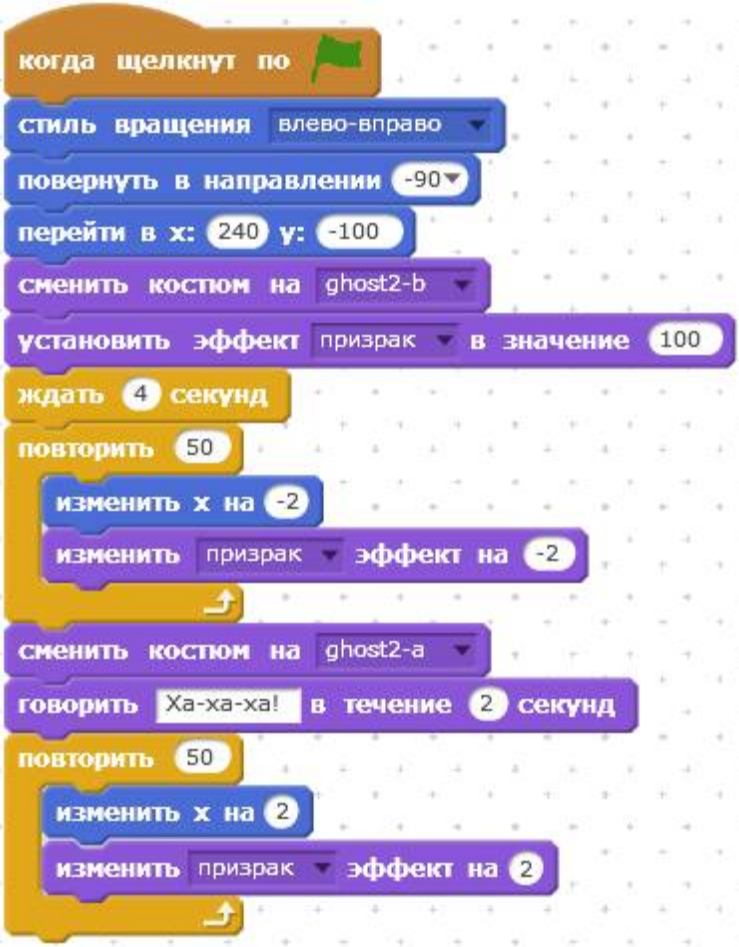
Теперь отцепите самые нижние два блока и удалите их, перетащив в область блоков.



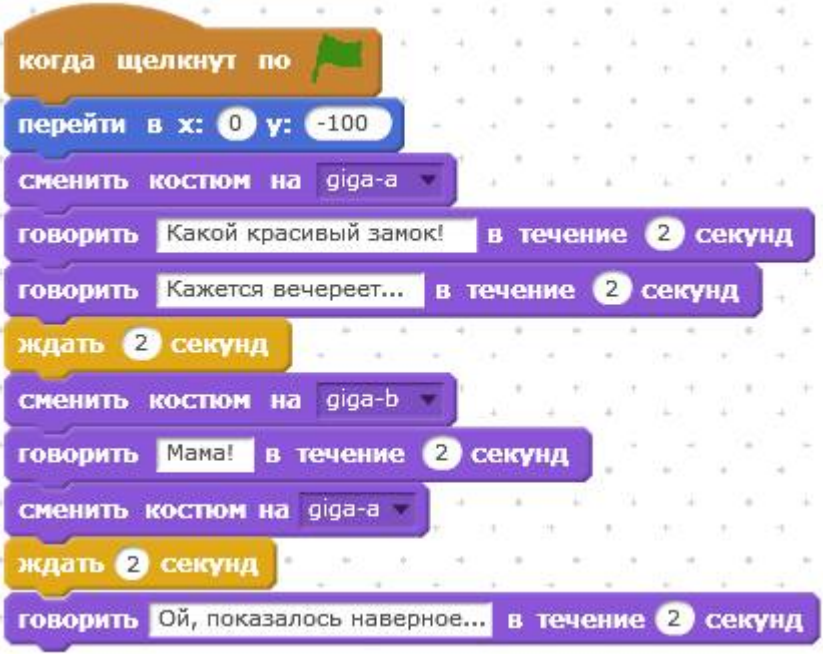
Осталось только изменить числа во втором блоке «повторить 50». X теперь должен изменяться на -2, так как Привидение будет уплывать влево, а эффект «призрак» должен изменяться на 2, чтобы Привидение плавно исчезло. В итоге получится вот такой скрипт первого Привидения.


когда щелкнут по
перейти в x: -240 y: -100
сменить костюм на ghost2-b
установить эффект призрак в значение 100
ждать 4 секунд
повторить 50
изменить x на 2
изменить призрак эффект на -2
сменить костюм на ghost2-a
говорить Ха-ха-ха! в течение 2 секунд
повторить 50
изменить x на -2
изменить призрак эффект на 2

Аналогичным способом измените скрипт второго Привидения. Он должен получиться вот таким.


когда щелкнут по
стиль вращения влево-вправо
повернуть в направлении -90
перейти в x: 240 y: -100
сменить костюм на ghost2-b
установить эффект призрак в значение 100
ждать 4 секунд
повторить 50
изменить x на -2
изменить призрак эффект на -2
сменить костюм на ghost2-a
говорить Ха-ха-ха! в течение 2 секунд
повторить 50
изменить x на 2
изменить призрак эффект на 2

Запустите проект. Привидения пугают Гигу и исчезают, но Гига так и осталась испуганной. Давайте вернём ей улыбку и хорошее настроение. Добавьте внизу к скрипту Гиги два соответствующих блока. Программа Гиги в итоге должна выглядеть так.


когда щелкнут по
перейти в x: 0 y: -100
сменить костюм на giga-a
говорить Какой красивый замок! в течение 2 секунд
говорить Кажется вечереет... в течение 2 секунд
ждать 2 секунд
сменить костюм на giga-b
говорить Мама! в течение 2 секунд
сменить костюм на giga-a
ждать 2 секунд
говорить Ой, показалось наверное... в течение 2 секунд

Проект готов. Не забудьте сохранить его.

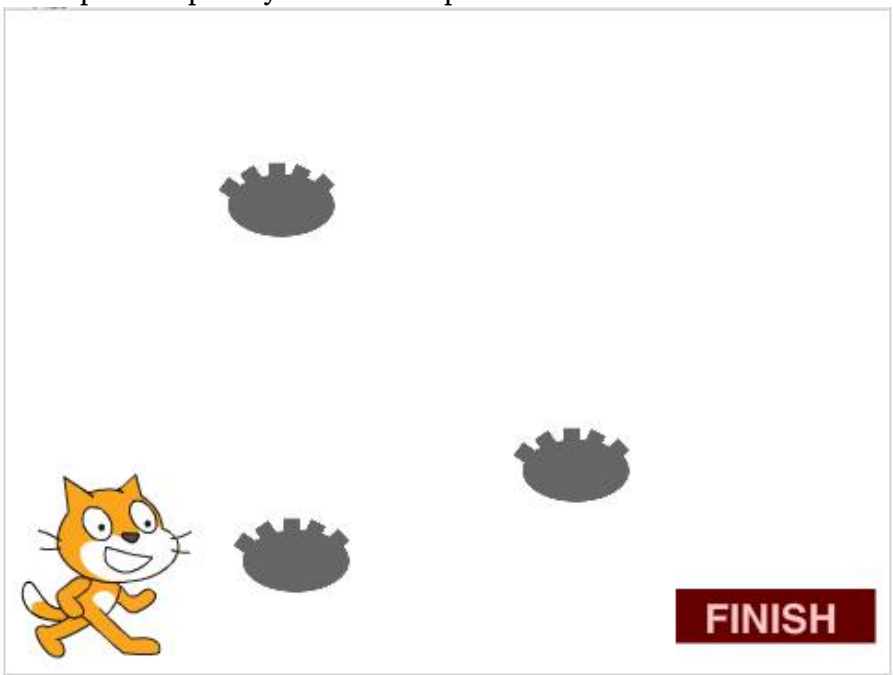
Вопрос. Как замедлить появление привидений?



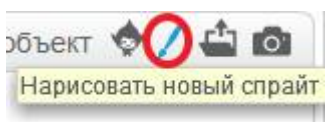
Задание. Добавьте в конце скрипта Гиги фразу «я пошла домой» и сделайте так, чтобы она уходила вправо до края экрана и исчезала. После того, как она исчезнет, яркость сцены должна измениться так, чтобы сцена стала совсем чёрной, как будто наступила ночь.

ГЛАВА 11. ИГРА «КОТЁНОК НА МИННОМ ПОЛЕ».

Как-то раз, Котёнок убежал от бабушки, и заблудился на военном полигоне где всё было заминировано. Задача Котёнка в этой игре — безопасно добраться до финиша. Если Котёнок наступит на мину, то он исчезнет, придётся начинать игру сначала. В этом проекте мы познакомимся с новыми блоками «таймер» и «перезапустить таймер».



Создайте новый проект. Первым делом нам надо нарисовать одну мину и Финиш. Нажмите на кнопку «Нарисовать новый спрайт».



Откроется редактор в битовом режиме. Выберите инструмент «Эллипс»



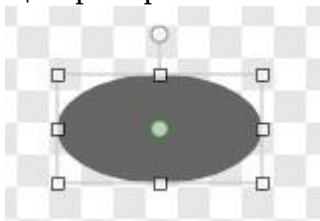
Выберите серый цвет.



Выберите закрашенный овал.



Нарисуйте овал в центре экрана.



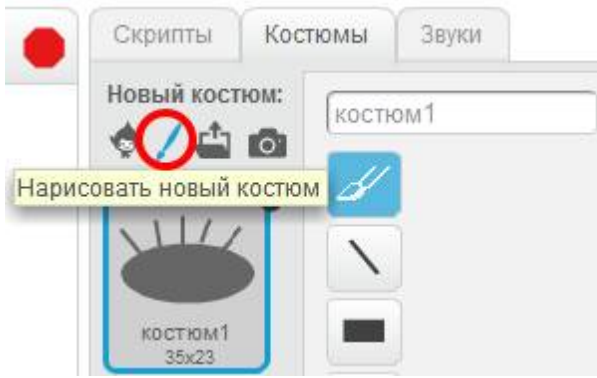
Затем выберите инструмент



Подрисуйте взрыватели.



У Мины будет два костюма, просто Мина и Взрыв. Первый костюм мы уже нарисовали, теперь надо нарисовать второй костюм. Нажмите на кнопку «Нарисовать новый костюм».



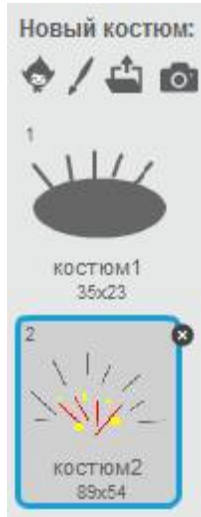
Выберите инструмент «Кисть».



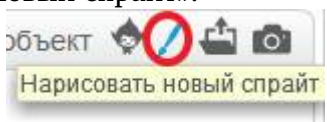
Нарисуйте разноцветный взрыв. В центре взрыва используйте жёлтый и красный цвет, а по краям серый и чёрный.



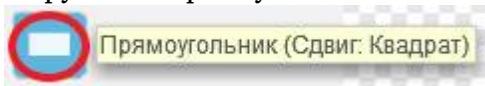
Теперь спрайт мины готов. У него два костюма.



Осталось нарисовать спрайт Финиш. Снова кликните на кнопку «Нарисовать новый спрайт».



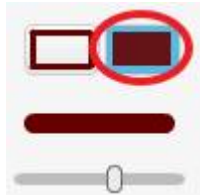
Выберите инструмент «Прямоугольник».



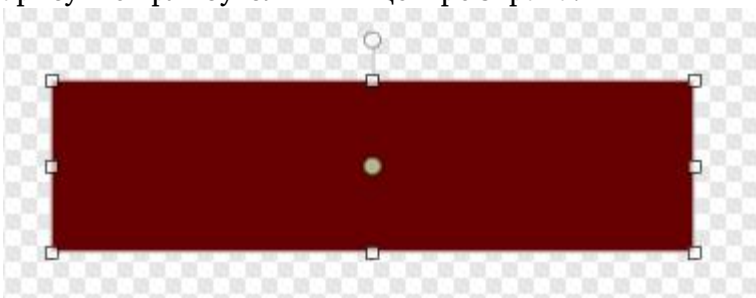
Выберите любой цвет, например, вот такой.



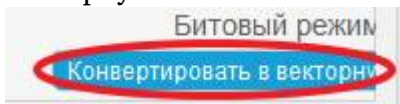
Выберите закрашенный прямоугольник.



Нарисуйте прямоугольник в центре экрана.



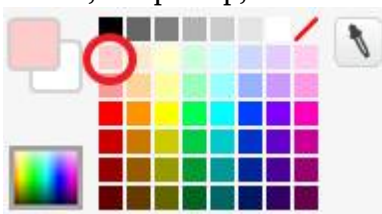
Сейчас надо написать слово FINISH. К сожалению, писать по-русски в Scratch 2 нельзя, поэтому напишем по-английски. Перейдите в векторный режим рисования. Нажмите на кнопку «Конвертировать в векторную».



Нажмите на кнопку «Текст».



Выберите цвет текста, например, такой.



Кликните в прямоугольник.



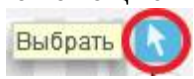
Напишите FINISH. Если ничего не пишется, значит, вы не переключили раскладку клавиатуры на русский язык. Переключите раскладку с помощью сочетания клавиш Ctrl + Shift или Alt + Shift.



Если буквы не помещаются в прямоугольник,



то их можно выделить с помощью инструмента «Выбрать».



А затем уменьшить размер, передвигая угловые квадратики. Также с помощью этого инструмента можно переместить слово в центр прямоугольника. При перемещении курсор станет похож на руку.



Все спрайты нарисованы. Можно начинать программировать. Сделайте такую программу для Кота.



Движение Кота управляется стрелками. Если нажата стрелка направо, то изменяем X на 1, то есть Кот двигается направо. Стрелка влево — меняем X на -1, Кот двигается влево. Стрелка вверх — меняем Y на 1, Кот двигается вверх. Стрелка вниз — меняем Y на -1, Кот двигается вниз. Всегда, если Кот коснётся Мины, которая называется Sprite2, то он подождёт одну секунду и исчезнет. Для Мины сделайте вот такую программу.



В начале программы Мина «одевает» костюм1.



Если Мина коснётся Кота, которого у нас зовут Sprite1, то она сменит костюм на костюм2, подождёт одну секунду и исчезнет.



Скрипт Финиша вот такой.



Здесь использованы два новых блока «перезапустить таймер» и «таймер». Таймер — это такой секундомер в Scratch. Когда мы перезапускаем его, то он начинает отсчёт секунд с нуля. Овальный блок «таймер» всегда содержит в себе количество секунд, которое прошло с момента запуска. Чтобы увидеть, как работает таймер, запустите программу зелёным флажком и дойдите Котом до таймера. В тот момент, когда Кот коснётся Финиша, над ним появится количество секунд, которое прошло с начала программы — это начнёт работать блок «сказать таймер».



Секунды будут бежать до тех пор, пока Кот касается Финиша. По правилам этой игры надо коснуться Финиша и отойти немного назад, тогда секунды перестанут бежать и остановятся на одном значении - это и будет Ваш результат.



Теперь протестируйте работу Мины. Запустите программу снова и подойдите Котом к Мине.

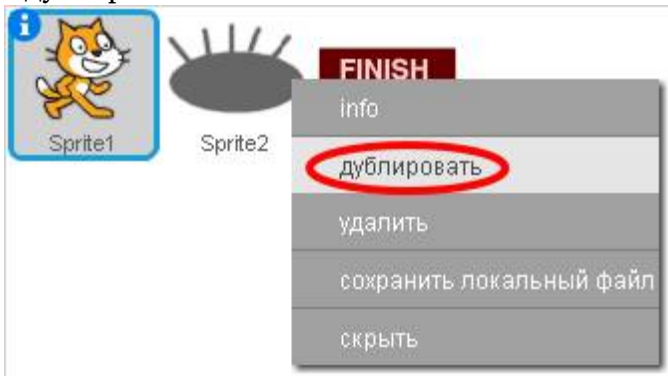


Как только Кот дотронется до Мины, будет взрыв.



Затем и Кот, и Мина исчезнут. Если всё работает правильно,

то можно усложнить игру, ведь играть с одной миной не интересно, её очень легко обойти. Для того чтобы очень просто добавить ещё одну мину, можно скопировать существующую. Для этого кликните на спрайт Мины правой кнопкой мышки и выберите «дублировать».



Появится ещё одна Мина — Sprite4. Самое главное — мина скопировалась вместе со скриптом! Продублируйте первую мину ещё раз.



Снова кликните на спрайт Мины правой кнопкой мышки и выберите «дублировать».



Появится третья Мина — Sprite5.



Разместите три мины на сцене так, чтобы затруднить Коту дорогу к Финишу.



Так как у нас теперь три мины, то программу Кота надо переделать, ведь он реагирует только на касание Sprite2. Самый простой способ научить Кота реагировать на новые мины, это дублировать часть скрипта. Выберите спрайт Кота, кликните правой кнопкой мышки на блок «если» и выберите дублировать.



Приклейте скопированный кусочек скрипта под блоком «если».

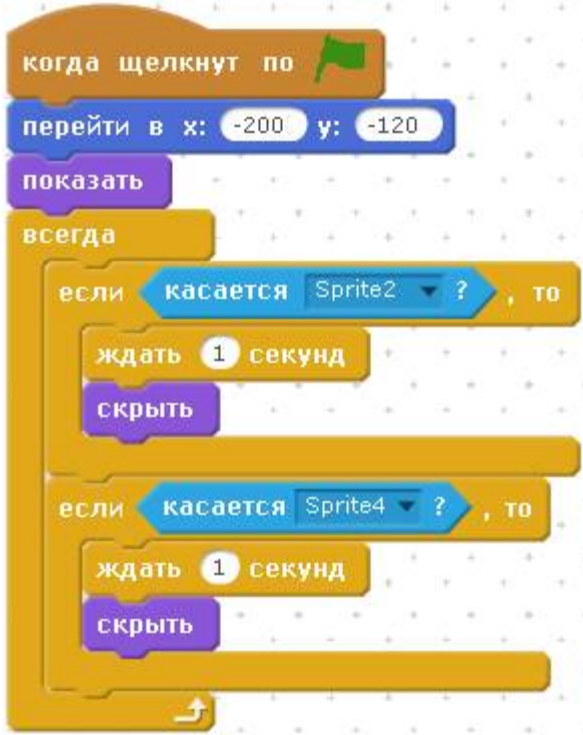


Получится вот так.

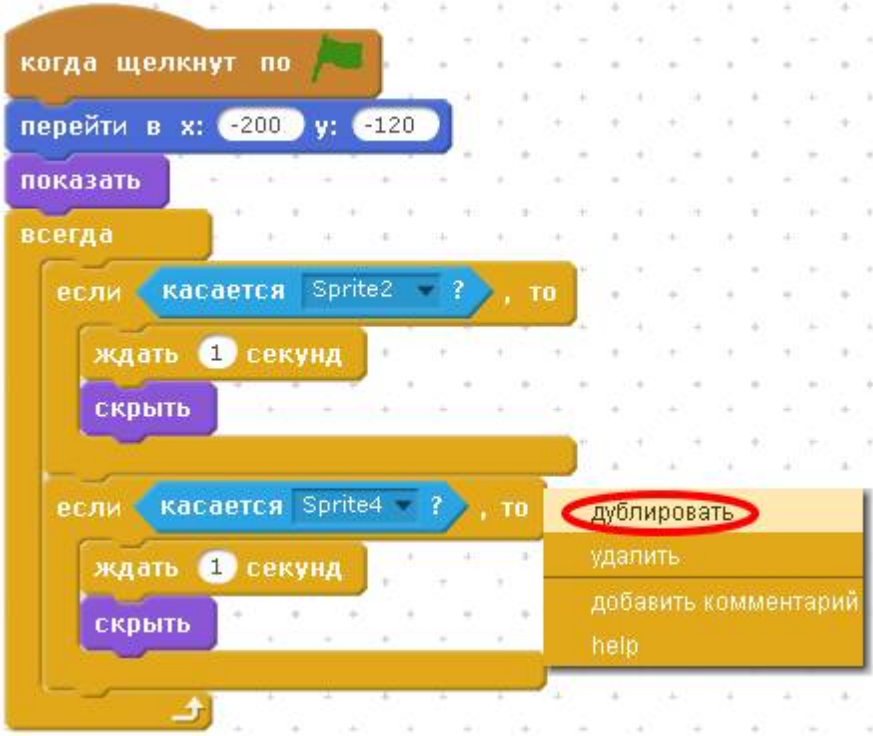


Измените во втором блоке «если» Sprite2 на Sprite4 — это имя

второй Мины.



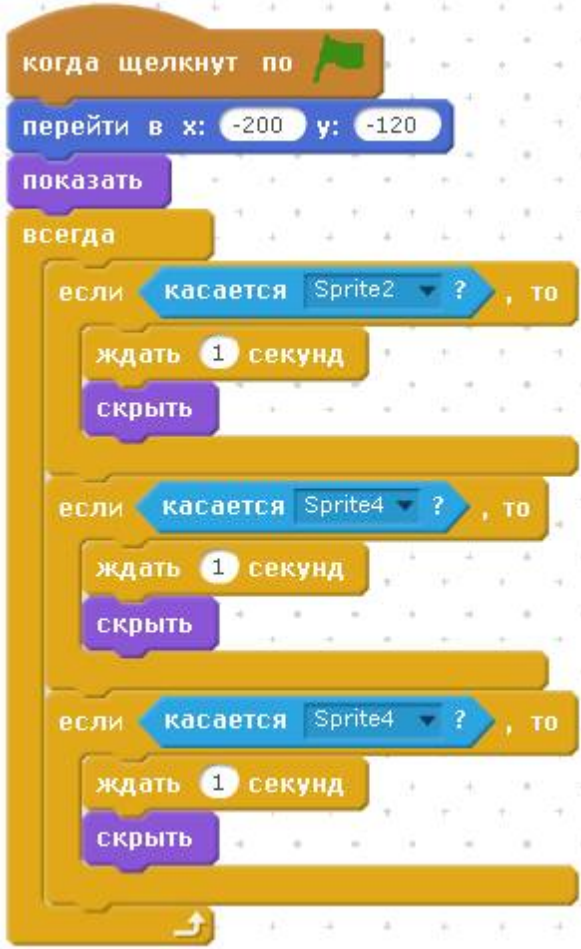
Теперь дублируйте второй блок «если».



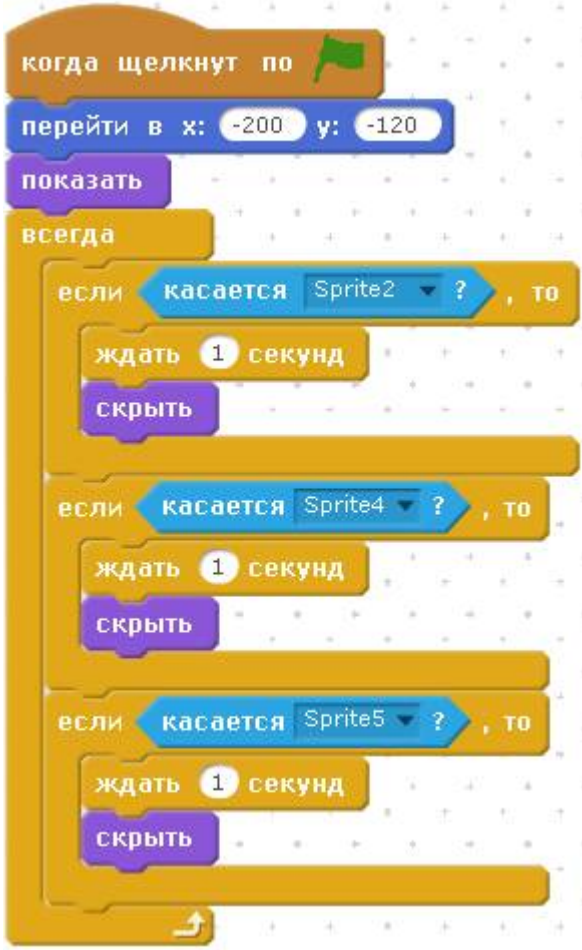
Приклейте копию второго блока «если» к нему снизу.



Должно получиться вот так.



Измените в третьем блоке «если» Sprite4 на Sprite5. В итоге этот спрайт Кота должен выглядеть вот так.



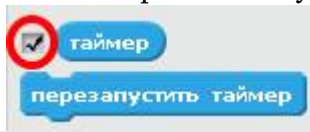
Запустите игру. Дойдите Котом до Финиша, не касаясь мин. Дотроньтесь до Финиша и сразу же отойдите. Мой результат больше 21 секунды, а у Вас?

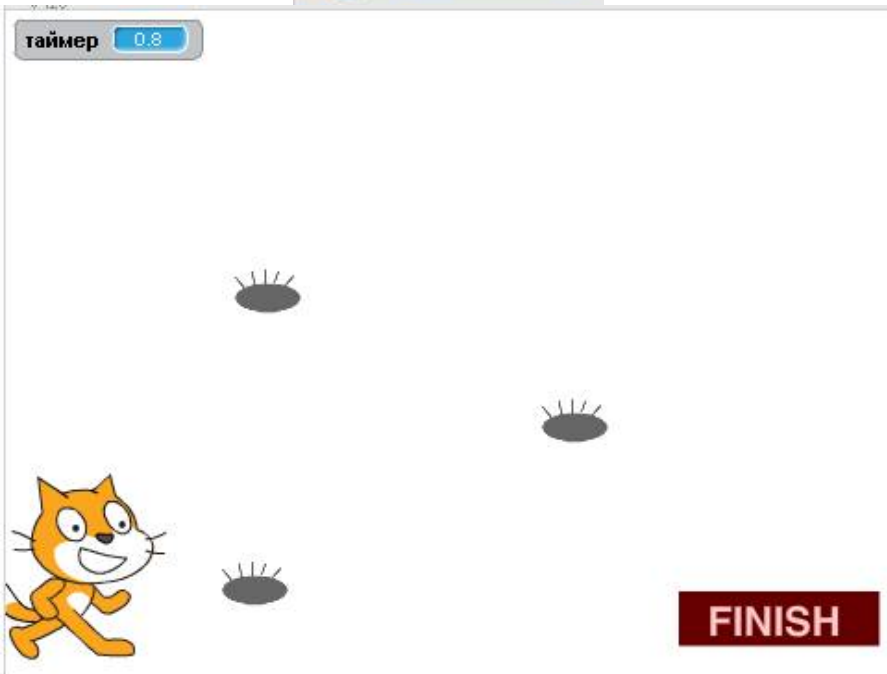


Пригласите друзей и устройте соревнования, кто пройдёт игру быстрее.



Совет. Можете поставить галочку напротив овального блока «таймер», тогда значение таймера можно будет видеть на экране.





Вопрос. Как ускорить передвижение Кота?



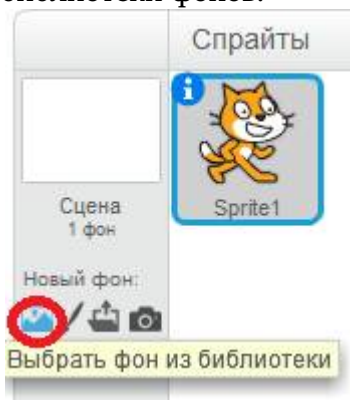
Задание. Для усложнения игры добавьте ещё одну мину.

ГЛАВА 12. ИГРА ПРО ВОЛШЕБНИКА.

У Волшебника старые счеты с Ведьмой. Как-то раз повстречались они в тёмном переулке, у Ведьмы с собой ничего не оказалось, а Волшебник имел при себе пять склянок с зельями. Кликая разные зелья, Волшебник накладывал на Ведьму различные магические заклинания. В этом проекте вы познакомитесь с сообщениями, а также потренируетесь рисовать в графическом редакторе.



Создайте новую программу. Сначала украсьте сцену красивым фоном из библиотеки фонов.



Выберите магический лес.



urban1



urban2

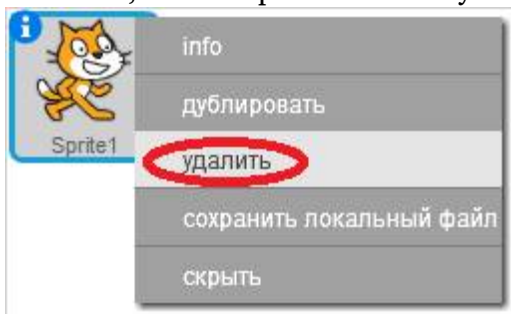


woods

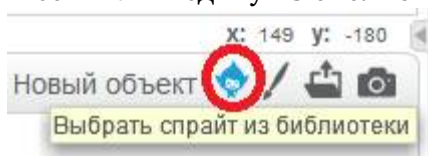


xy-grid

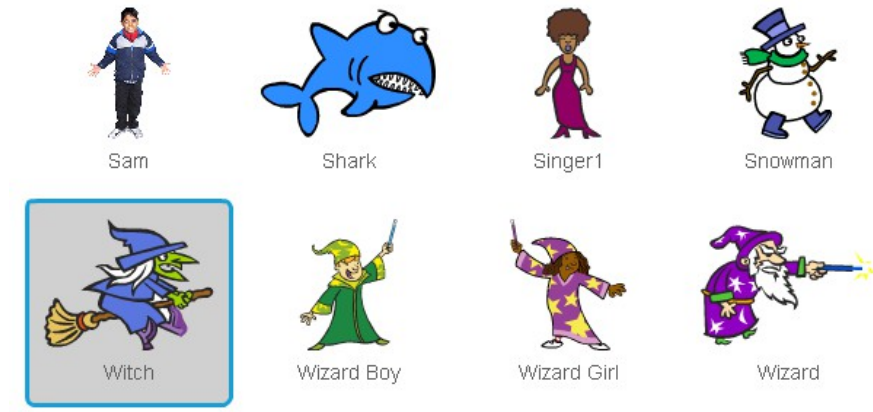
Затем удалите Кота, в этом проекте он не нужен.

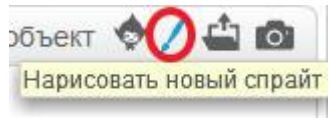


Загрузите Волшебника и Ведьму из библиотеки спрайтов.

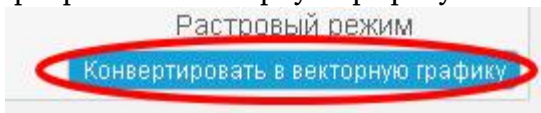


Склянок с волшебными зельями в библиотеке спрайтов нет. Нарисуем их сами. Кликните кнопку «Нарисовать новый спрайт».

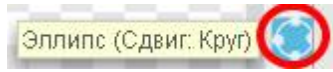




Рисовать склянки лучше в векторном режиме. Кликните кнопку «Конвертировать в векторную графику».



Выберите инструмент «Эллипс».



Выберите чёрный цвет.



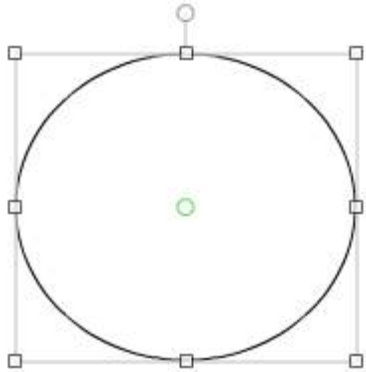
Выберите Тонкую линию и незакрашенный овал.



Увеличьте масштаб, кликните два раза на плюсик.



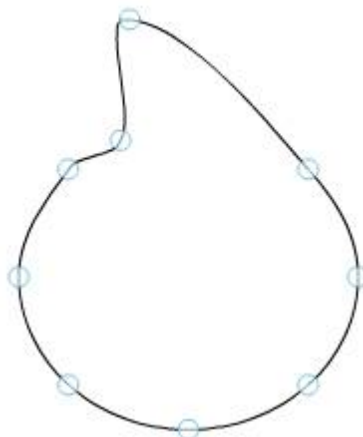
Нарисуйте круг или овал.



Выберите инструмент «Изменение формы».



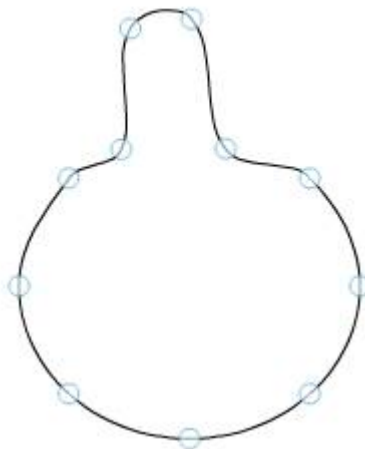
Кликните на чёрную линию между двумя точками привязки, в этом месте автоматически появится новая точка привязки. Перетащите её повыше, чтобы получилось горлышко склянки.



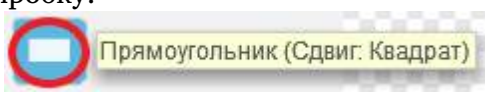
Снова кликните в чёрную линию и сделайте ещё одну точку.



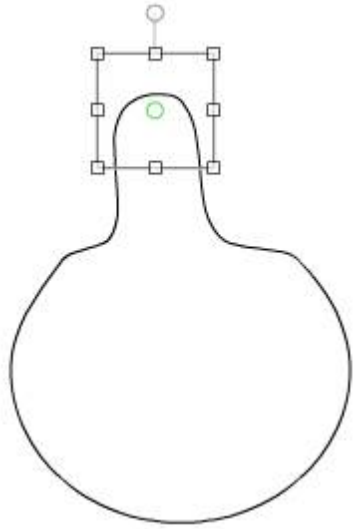
А затем ещё одну. Склянка готова.



Теперь выберите инструмент «Прямоугольник». С помощью него нарисуем пробку.

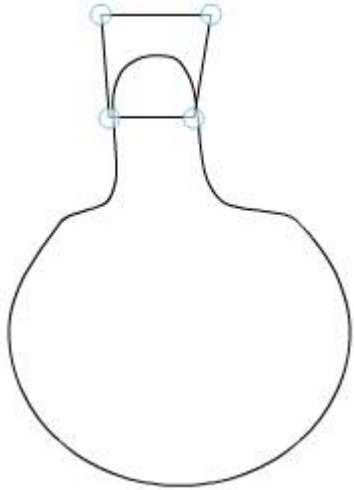


Нарисуйте прямоугольник.

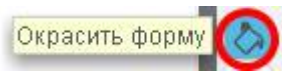


Затем выберите инструмент «Изменение формы» и немного передвиньте точки привязки.

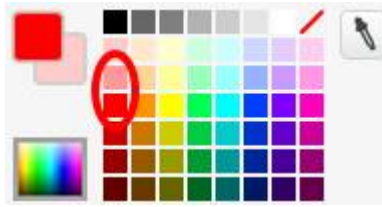




Осталось раскрасить склянку. Выберите инструмент «Окрасить форму».



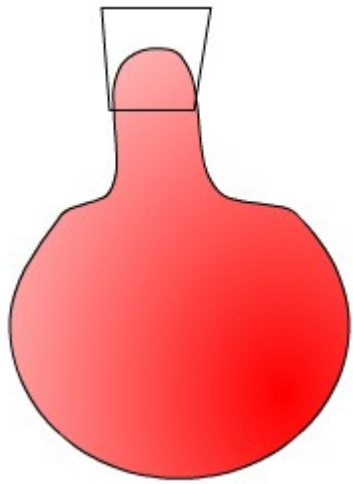
Выберите два цвета, красный и розовый.



И вот такой тип заливки.



Кликните на склянку справа внизу, там, где самый яркий цвет.



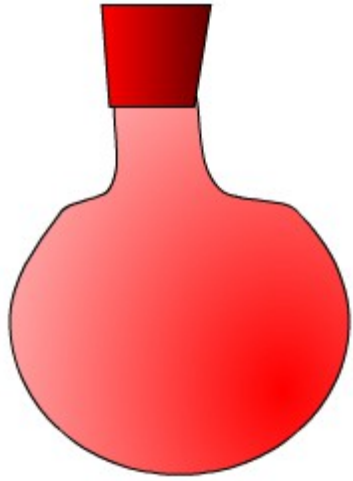
Теперь раскрасим пробку. Выберите такие цвета.



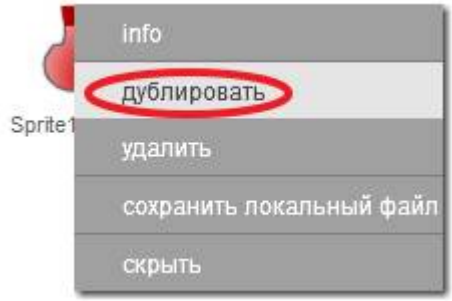
Выберите такой тип заливки.



Кликните в середину пробки. Получится вот такая склянка с красным зельем.



Скорее всего, склянка получит имя Sprite1. Дублируйте её и закрасьте вторую склянку зелёным цветом. Как раскрашивать, Вы уже знаете.



Снова дублируйте склянку и раскрасьте фиолетовым цветом. В итоге должно получиться пять разноцветных склянок.



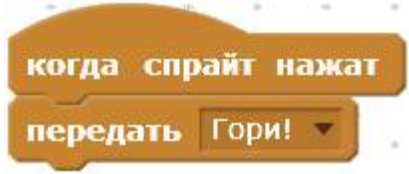
Теперь, когда все спрайты готовы, можно начинать программировать. Сделайте такой скрипт для красной склянки.



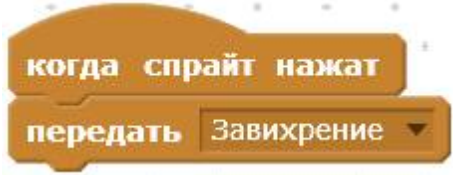
Когда Вы возьмёте блок «передать» в нём будет только одно сообщение «message1». Кликните на чёрный треугольничек и выберите «новое сообщение».



Здесь использован новый блок «передать». Этот блок передаёт сообщение под названием «цвет». Это сообщение может получить любой спрайт, и выполнить какие-либо действия. Например, в нашем проекте Ведьма, получив сообщение «цвет», изменится при помощи эффекта «изменить цвет». Каждая склянка будет передавать своё сообщение, и Ведьма будет реагировать на каждое по-разному. Следующий скрипт для зелёной склянки.



Следующий скрипт для фиолетовой склянки.



Следующий скрипт для жёлтой.



Следующий скрипт для голубой.



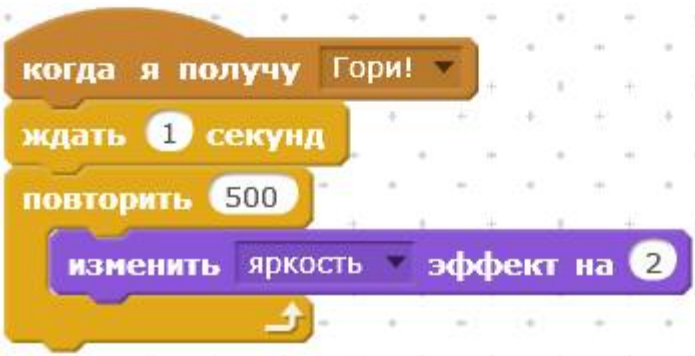
Следующая программа для Волшебника. Как видите, при поступлении каждого сообщения он говорит разную фразу.


когда щелкнут по
перейти в x: -100 y: -100
когда я получу Вздуйся
говорить Лопни, противная! в течение 2 секунд
когда я получу Гори!
говорить Гори, ведьма! в течение 2 секунд
когда я получу Завихрение
говорить Кружись, пока не развалишься! в течение 2 секунд
когда я получу Окаменей
говорить Окаменей! в течение 2 секунд
когда я получу Цвет
говорить Выпей компотика! в течение 2 секунд

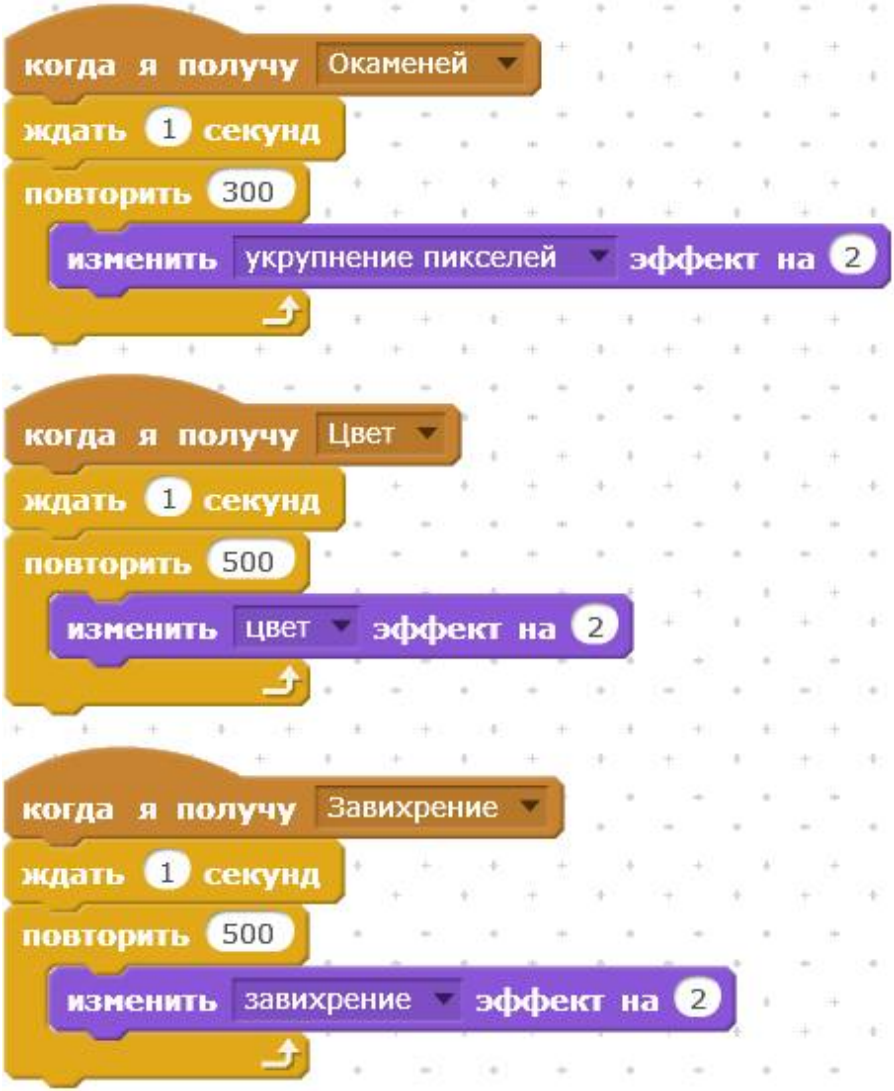
Следующая программа для Ведьмы. В ней тоже шесть скриптов.


когда щелкнут по
стиль вращения влево-вправо
перейти в x: 160 y: -100
повернуть в направлении -90
говорить Привет, дедуля! в течение 2 секунд





И ещё три скрипта для Ведьмы.



Как видите, при получения каждого сообщения Ведьма ждёт одну секунду, а затем начинает плавно изменяться с использованием различных эффектов.



Запустите проект и кликайте на склянки, кажется, Ведьме стало нехорошо.

Вопрос. Какие два заклинания наложены на ведьму?



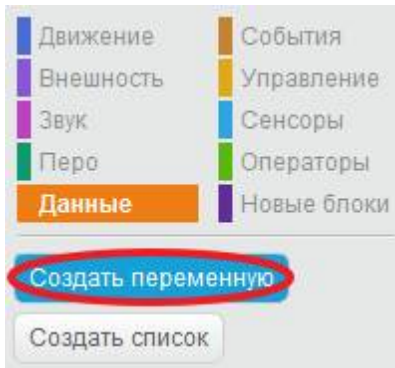
Задание. Помогите Ведьме. Нарисуйте для неё несколько колбочек, с помощью которых она сможет отомстить Волшебнику.



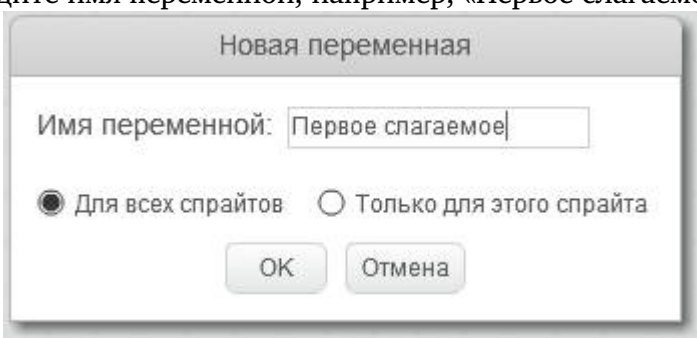
Подсказка. Использовать готовые сообщения нельзя. Придумайте новые сообщения.

ГЛАВА 13. КОТ МАТЕМАТИК.

Кот ходил в школу, очень хорошо учился и стал математиком. Теперь он может сложить два любых числа. В этой главе мы познакомимся с переменными. Переменные — это специальные ячейки в компьютерной памяти в которых можно хранить разные числа и слова. Название «переменная» означает что содержимое этой ячейки не постоянно и может изменяться. У каждой переменной есть своё имя. Чтобы создать переменную нажмите на оранжевые блоки «Данные» и на кнопку «Создать переменную».

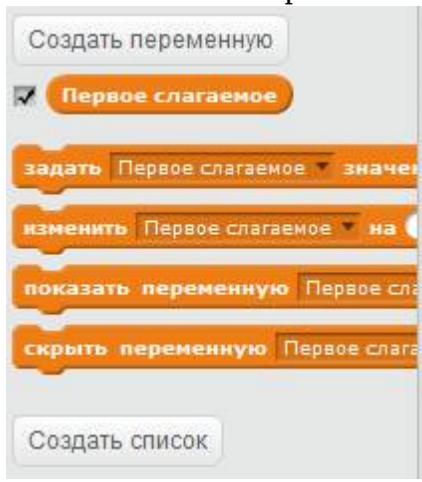


Введите имя переменной, например, «Первое слагаемое».



Имена переменным следует давать такие, чтобы было понятно, что в ней храниться. Если в банку с надписью «Сахар» Вы положите «Соль», и забудете написать на банке об этом, то Ваша Мама будет очень недовольна, когда будет пить солёный чай. Я боюсь даже представить, что с Вами будет, если Папа воспользуется баночкой с надписью «Соль», в которой хранился перец, и «посолит» супчик. Для того, чтобы в Вашей программе не возникало таких опасных ситуаций, я всем советую всегда давать переменным такие имена, чтобы было понятно, что в них храниться. В нашей программе мы будем складывать два числа, значит переменным нужно дать имена «Первое слагаемое» и

«Второе слагаемое». В тот момент, когда Вы создадите первую переменную, появятся новые блоки переменных.



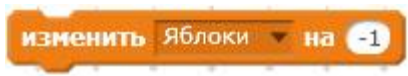
Вытащите их в область скриптов чтобы разглядеть получше.



Первый блок позваляет положить какое либо значение в переменную, или, другими словами, задать значение переменной. Второй блок позволяет изменить значение переменной на какою либо величину, то есть прибавить число к числу, хранящемуся в переменной. Например, если у спрайта «Ваня» есть переменная «Яблоки», то после выполнения блока «задать Яблоки значение 2» у Вани будет два яблока.

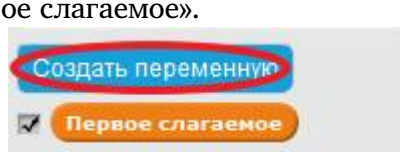


Если теперь выполнить блок «изменить Яблоки на -1».

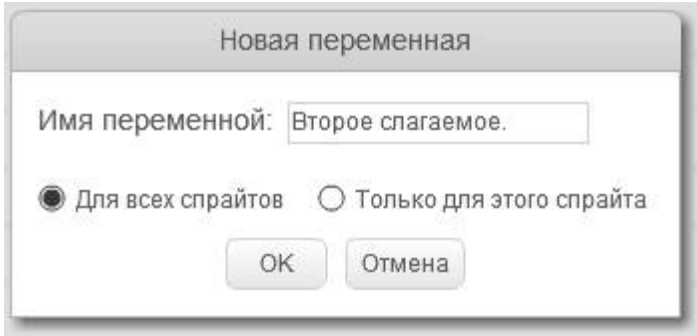


То у Вани останется только одно яблоко. Значение переменной «Яблоки» станет равно единице. Наверное Ваня слопал яблочко.

Продолжаем работу над проектом. Создайте следующую переменную «Второе слагаемое».



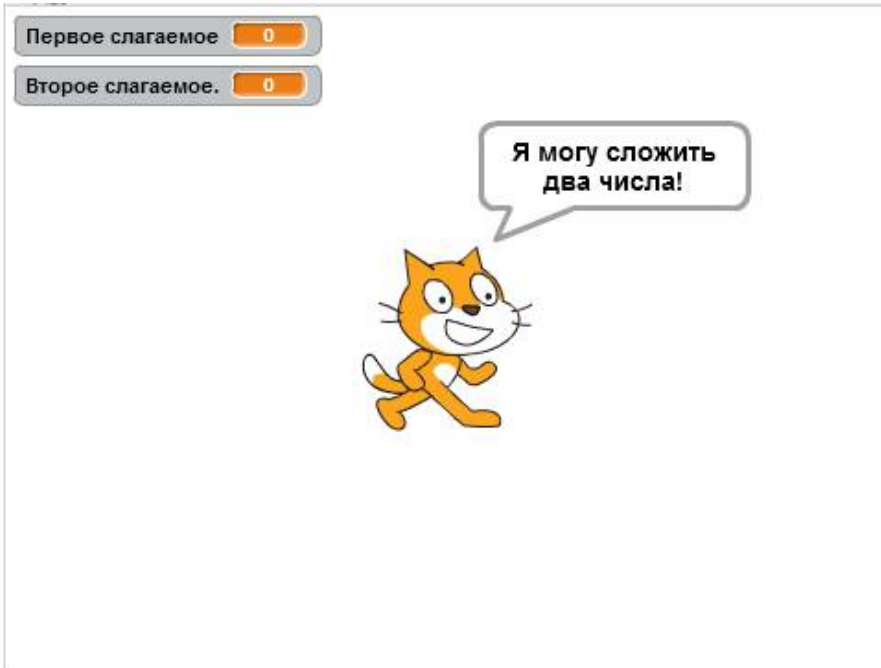
Введите имя переменной.



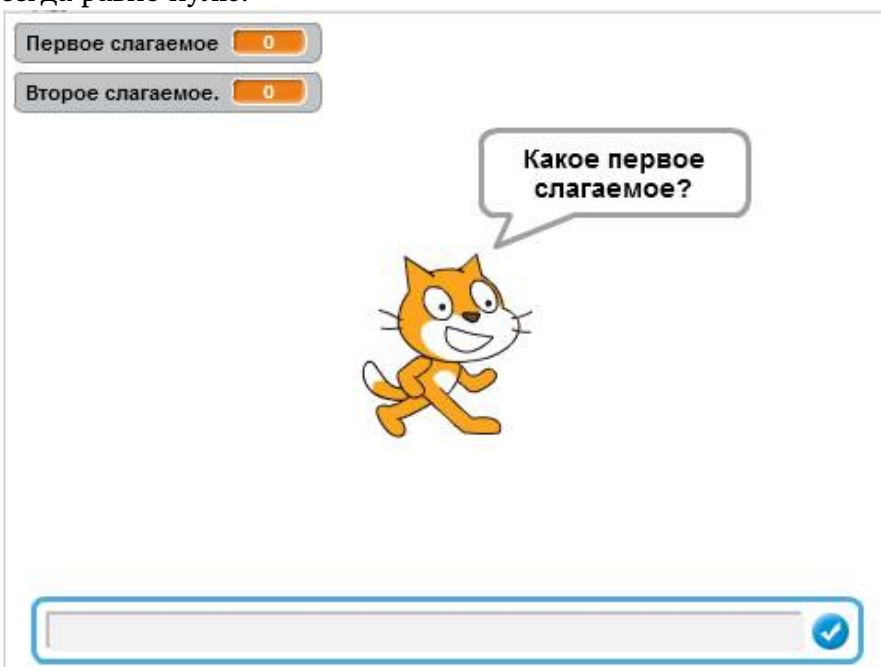
Теперь давайте научим Кота складывать числа. Сделайте ему такую программу. Зелёный блок сложения найдёте в блоках «операторы». Голубые блоки «спросить» и «ответ» найдёте в блоках «сенсоры».


когда щелкнут по
говорить Я могу сложить два числа! в течение 2 секунд
спросить Какое первое слагаемое? и ждать
задать Первое слагаемое значение ответ
спросить Какое второе слагаемое? и ждать
задать Второе слагаемое. значение ответ
сказать Первое слагаемое + Второе слагаемое.

Для того, чтобы сразу понять, как работают голубые блоки «спросить» и «ответ», запустите программу зелёным флажком.



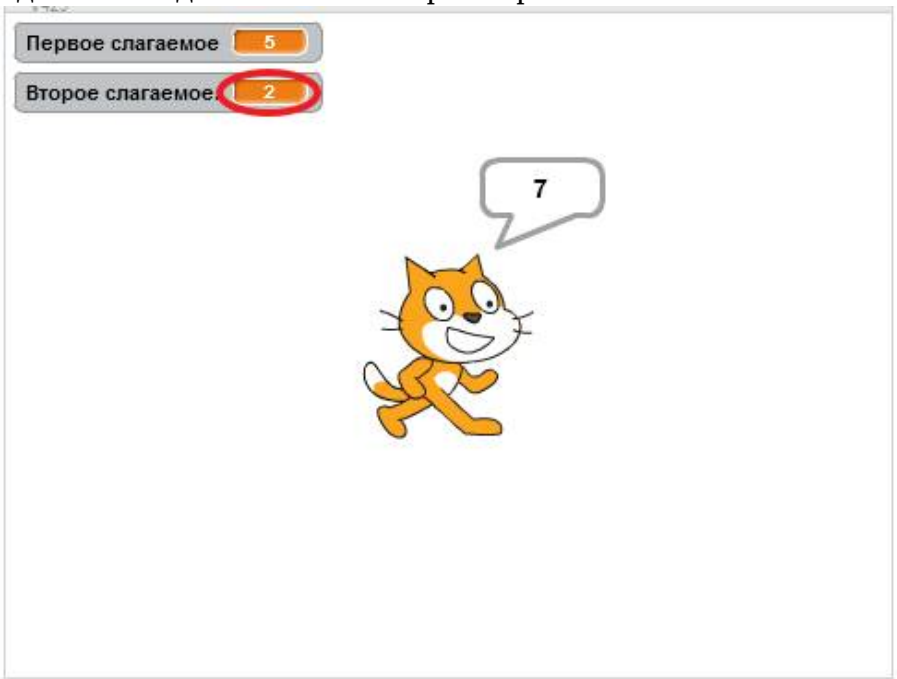
Обратите внимание, значения переменных отображены на экране. В начале, когда переменные только созданы, их значение всегда равно нулю.



При выполнении голубого блока «спросить» на экране отображается поле ввода данных. Напишите туда первое слагаемое и нажмите Enter.



Кот попросит ввести значение второго слагаемого, а пятёрочка тем временем запрыгнула в уютненькую переменную и отдыхает. Введите значение второй переменной и нажмите Enter.



Двойка запрыгнула во вторую переменную, а мудрый Котик сказал, что сумма двух введённых чисел, которые теперь хранятся в памяти компьютера, равна семи. Значения в переменных могут

храниться очень долго, до тех пор, пока вы не измените их с помощью блоков «задать» или «изменить».

Запустите программу снова.



Обратите внимание, из-за того, что мы нажали на зелёный флажок, значения переменных не изменились. Пятёрка и Двойка прекрасно себя чувствуют. Однако, теперь Котик попросил ввести новое значение первого слагаемого. Введите десять и нажмите Enter.

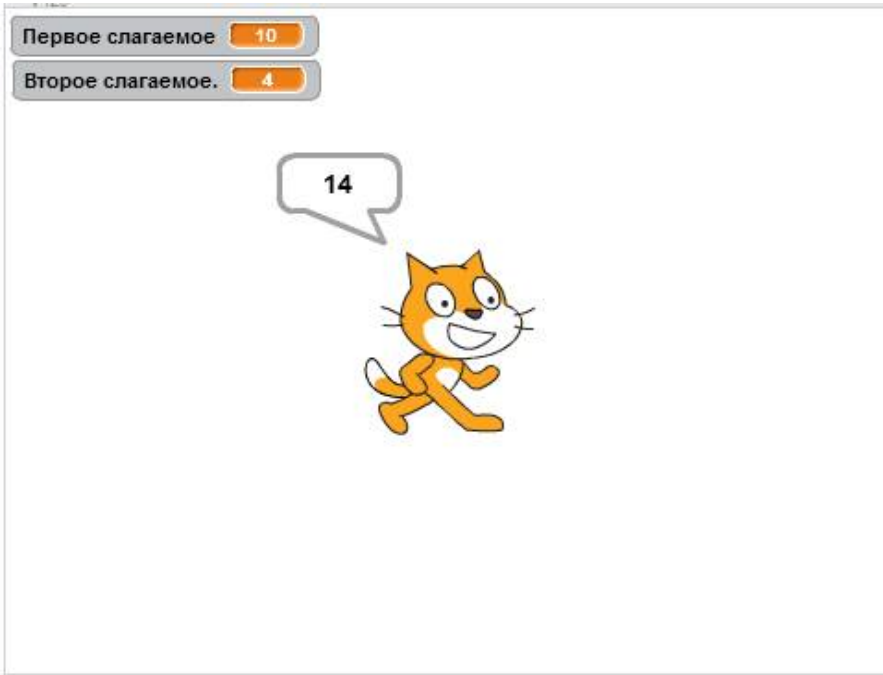


Значение первого слагаемого изменилось, теперь там храниться десяточка. Введите значение второго слагаемого, например, четыре.



Переменная «Второе слагаемое» теперь бережно хранит

четвёрку. А Котик продолжает радовать нас своими математическими способностями.



Вопрос. Сколько будет два рубля плюс тысяча копеек?



Задание. Научите Кота умножению.

ГЛАВА 14. ИГРА «КОТ С РЕАКТИВНЫМ РАНЦЕМ».

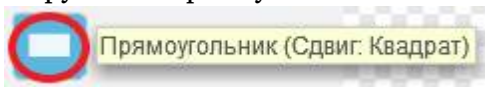
Как-то раз гулял Котик по лесу и забрёл на делянку где велась заготовка леса. Прораб был очень недоволен и кидался брёвнами! Пришлось Коту вспомнить о своём реактивном ранце, если бы не он, Коту несдобровать. В этой главе мы сделаем небольшую игру про Котика, который будет летать на реактивном ранце, уворачиваясь от летящих брёвен. Мы познакомимся с блоком «выдать случайное число», а также попрактикуемся в применении переменных. Как обычно, будем осваивать новые приёмы рисования. Игра будет выглядеть вот так.



Создание проекта начнём с фона. Нажмите на кнопку «Нарисовать новый фон».



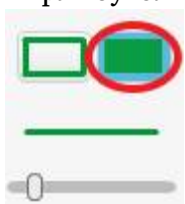
Выберите инструмент «Прямоугольник».



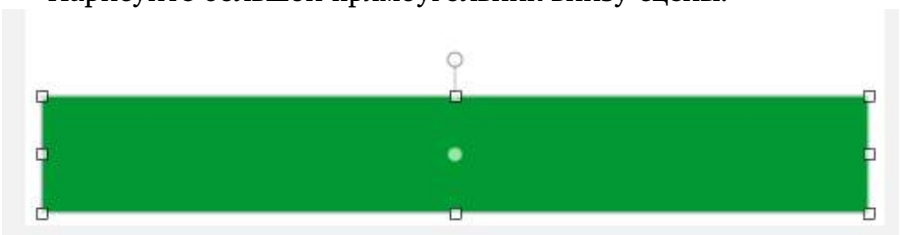
Выберите зелёный цвет.



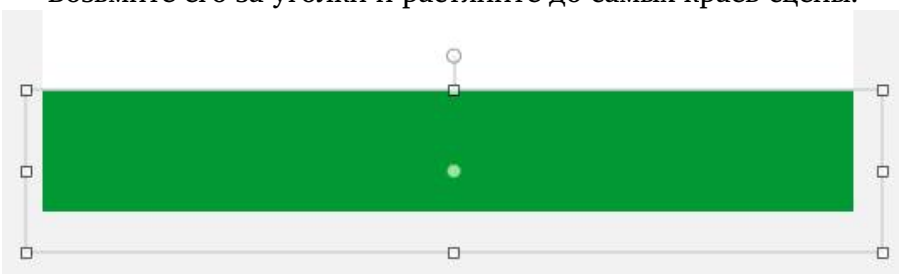
Выберите закрашенный прямоугольник.



Нарисуйте большой прямоугольник внизу сцены.

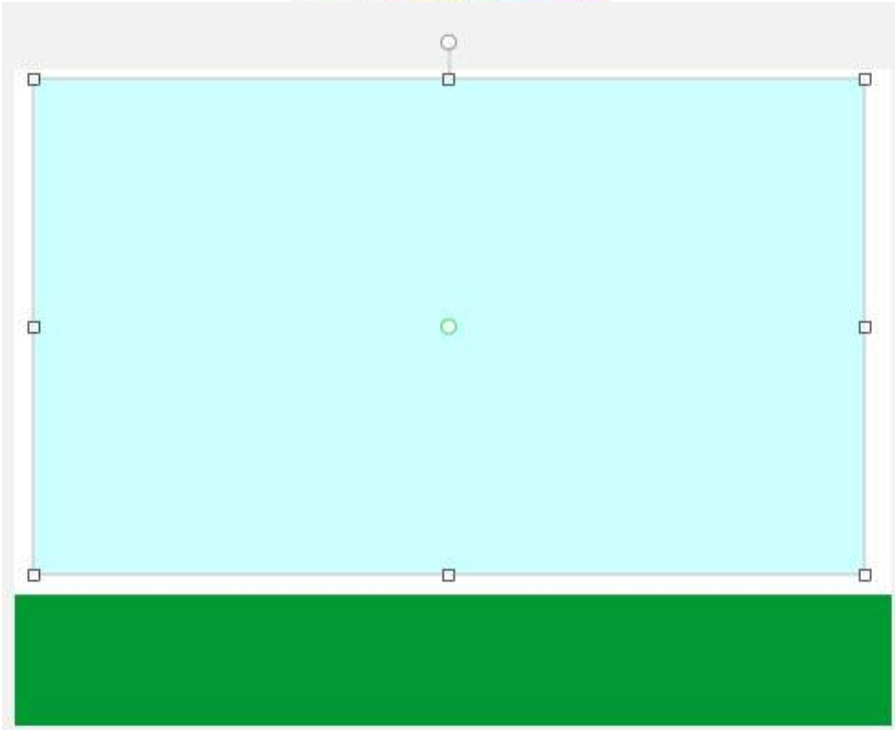


Возьмите его за уголки и растяните до самых краёв сцены.

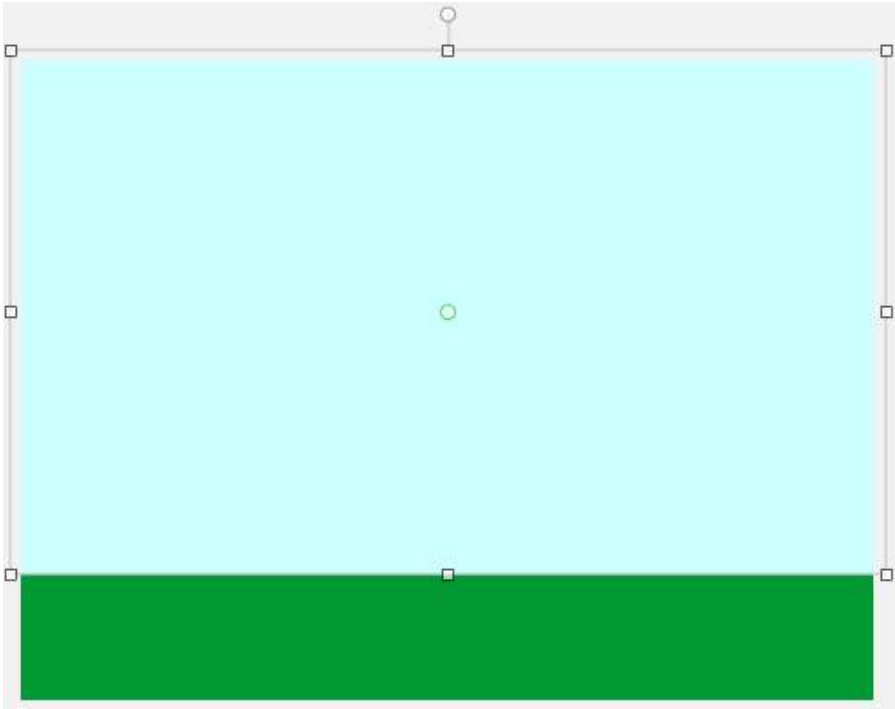


Теперь выберите голубой цвет и нарисуйте большой прямоугольник на всю оставшуюся сцену.

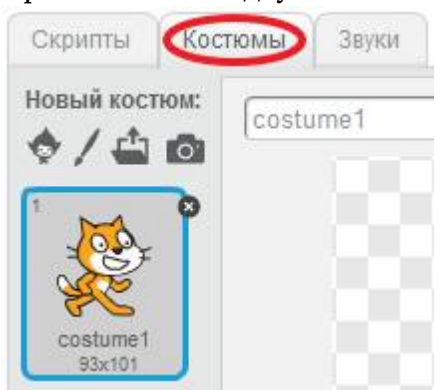




Растяните прямоугольник так, чтобы не осталось пустого белого места.



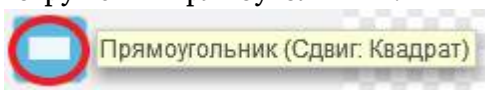
Фон готов. Нарисуем Коту реактивный ранец. Выберите Кота и откройте редактор нажав на вкладку «Костюмы».



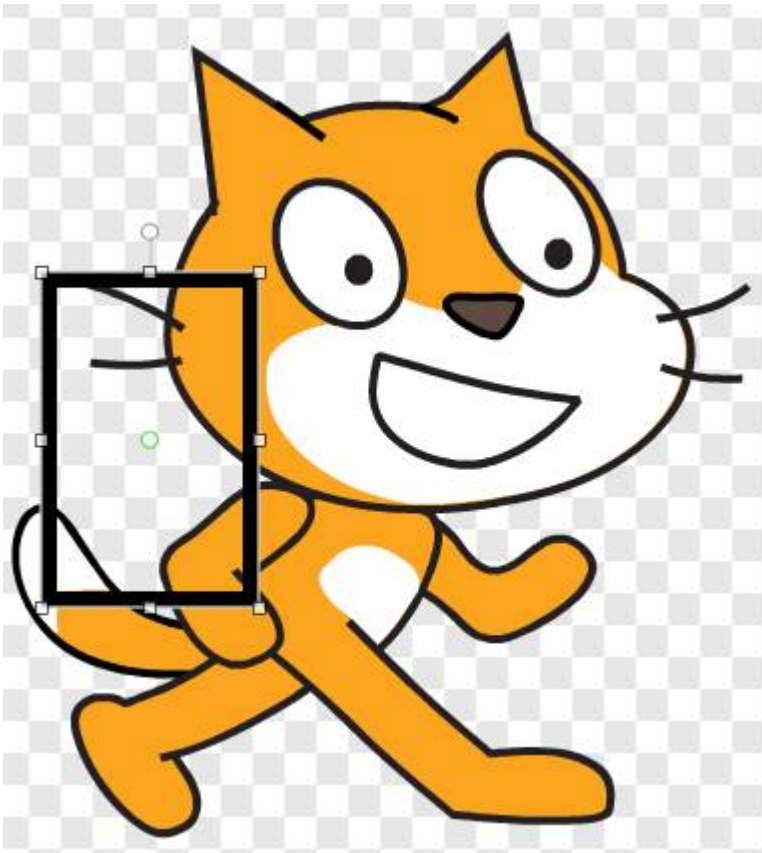
Увеличьте масштаб.



Выберите инструмент «Прямоугольник».



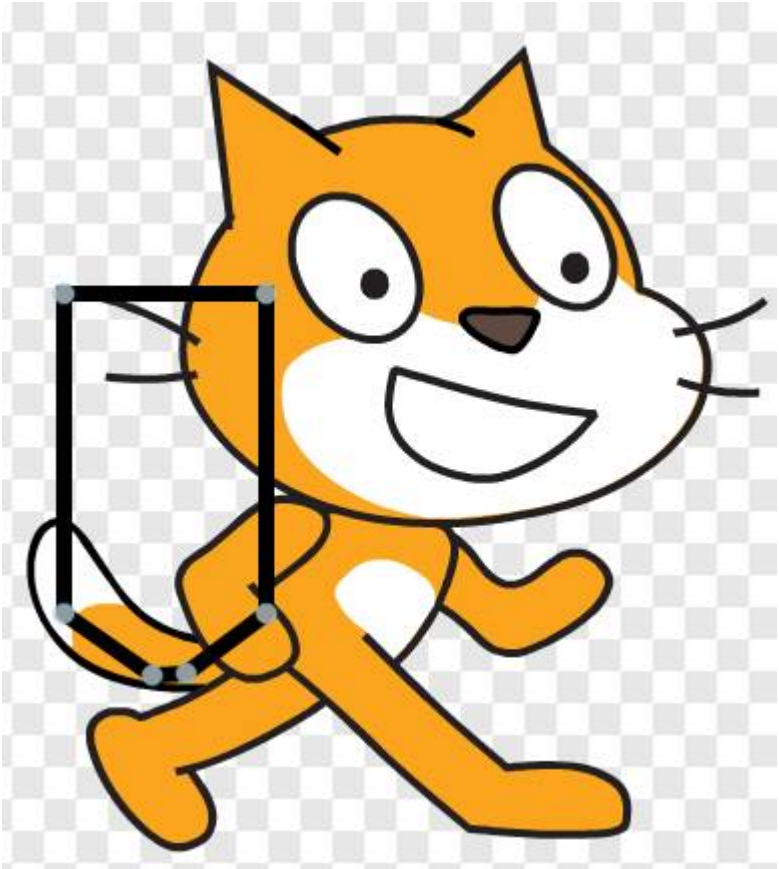
Рисуйте чёрным цветом.



Выберите инструмент «Изменение формы».



Кликните на нижний край прямоугольника и измените его форму.



Теперь надо закрасить реактивный ранец. Выберите инструмент «Заполнить цветом».



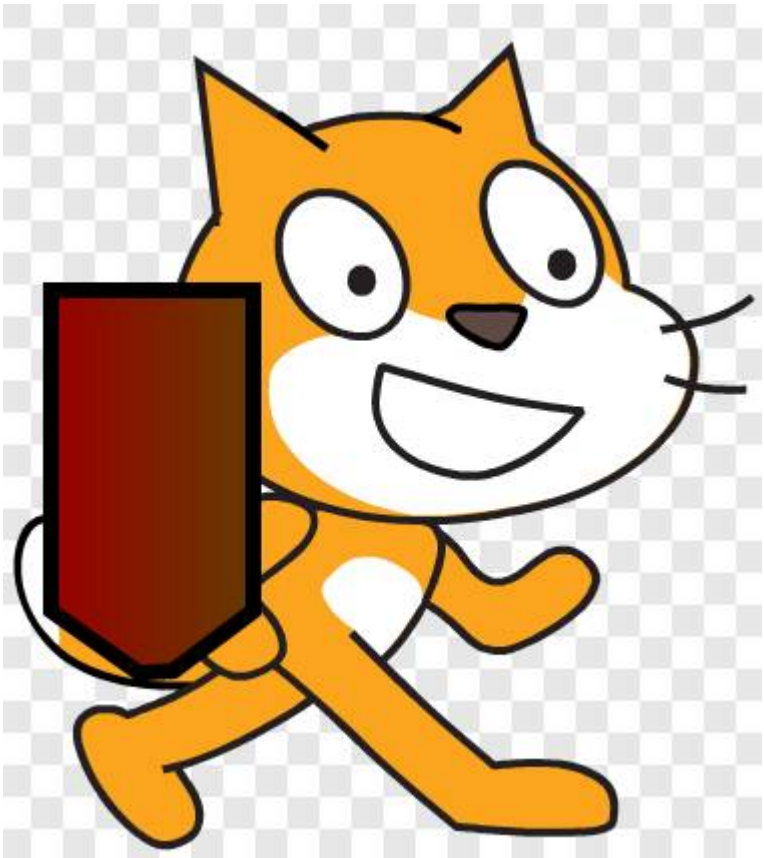
Выберите два цвета.



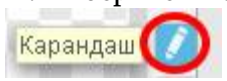
Выберите тип заливки.



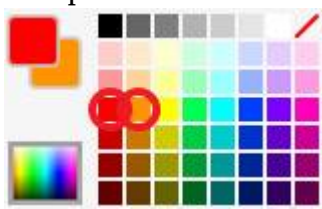
Закрасьте ранец.



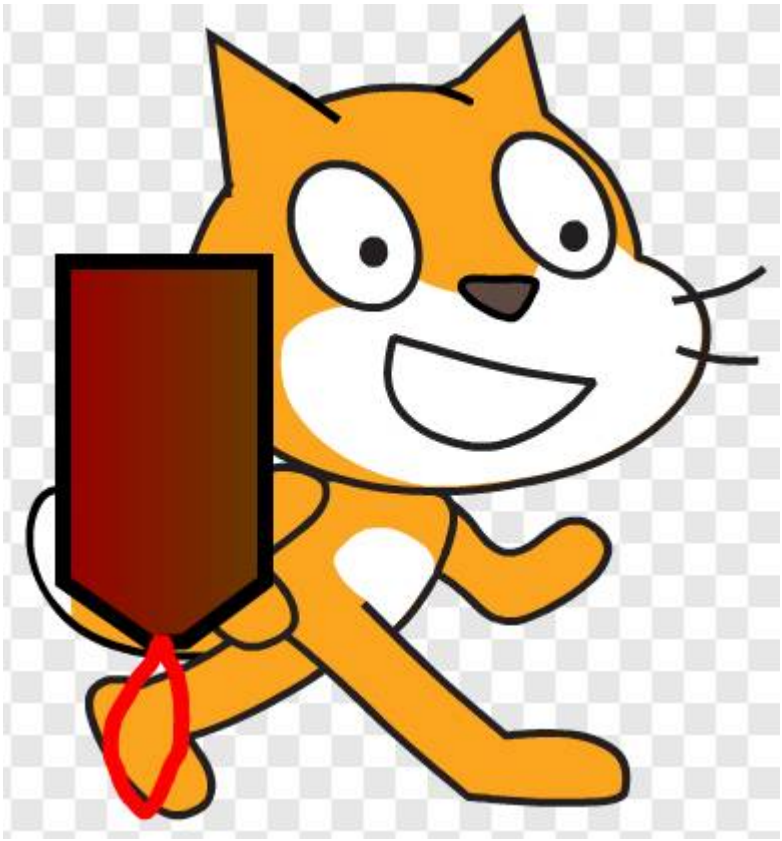
Теперь нарисуем огонь. Выберите инструмент «Карандаш».



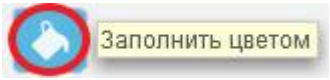
Выберите красный и оранжевый.



Нарисуйте огонь в виде замкнутого контура. Это нужно для того, чтобы потом его можно было окрасить.



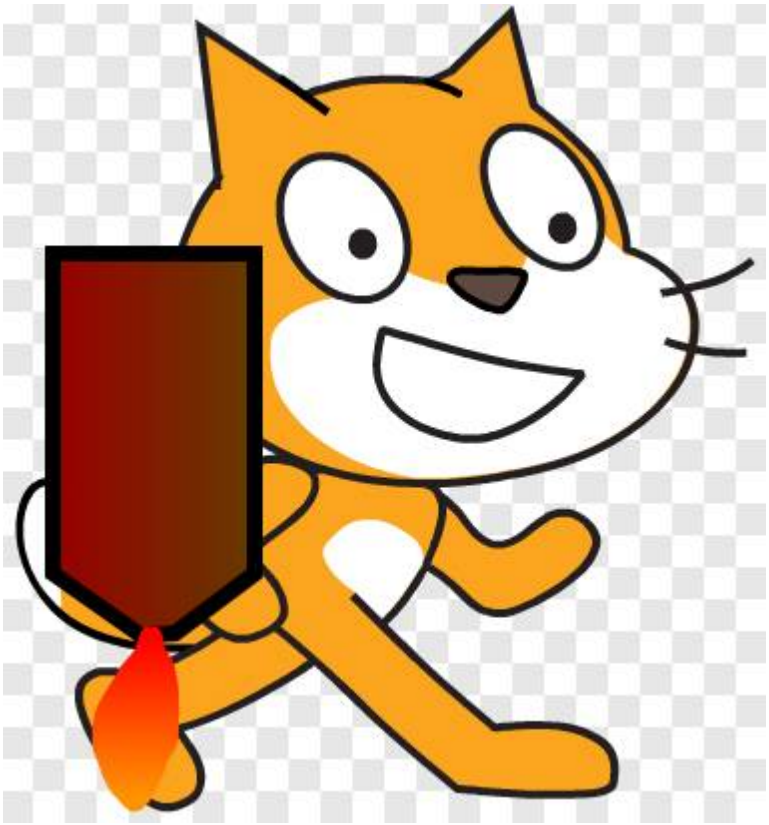
Выберите инструмент «Заполнить цветом».



Выберите способ заливки.



Окрасьте огонь и его контур.



Теперь надо немного наклонить рюкзак. Кликните в него инструментом выбора.



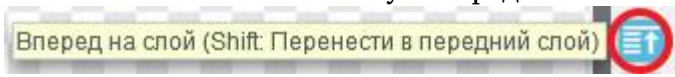
Потяните рюкзак за верхний кружочек и наклоните его.



Теперь выберите огонь, наклоните его и переместите на своё место.



Осталось переместить Кота на уровень вверх, чтобы реактивный ранец остался у него за спиной. Инструментом выбора кликните на Кота и на кнопку «Вперёд на слой».



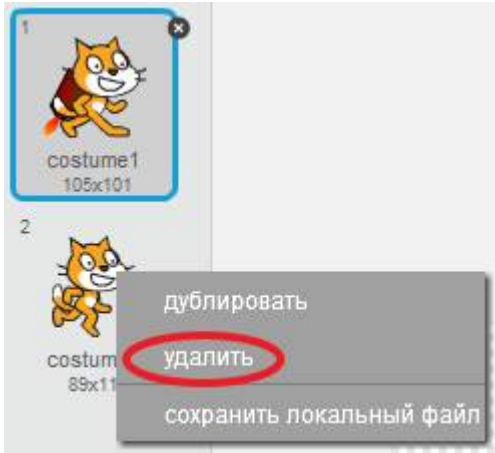
Котик готов к полётам!



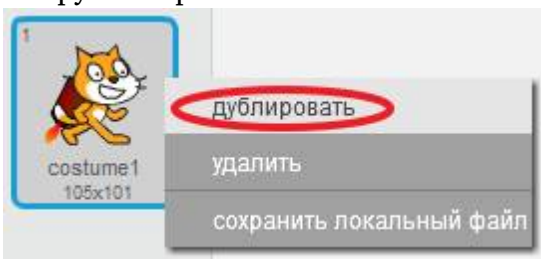
Теперь надо нарисовать ему грустный костюм, таким он станет при попадании в него бревна. Перейдите на вкладку «Костюмы».



Сначала удалим второй костюм Котика.



Теперь дублируйте первый костюм.



Увеличьте масштаб. Дважды кликните на плюс.



Выберите инструмент «Изменение формы».



Кликните на рот и передвиньте вних левую и правую точки привязки. Рот стал грустным.



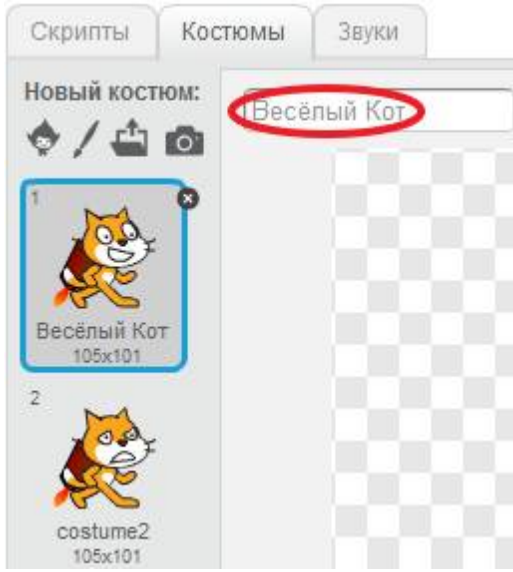
Теперь кликните на глаз и передвиньте верхнюю точку пониже.



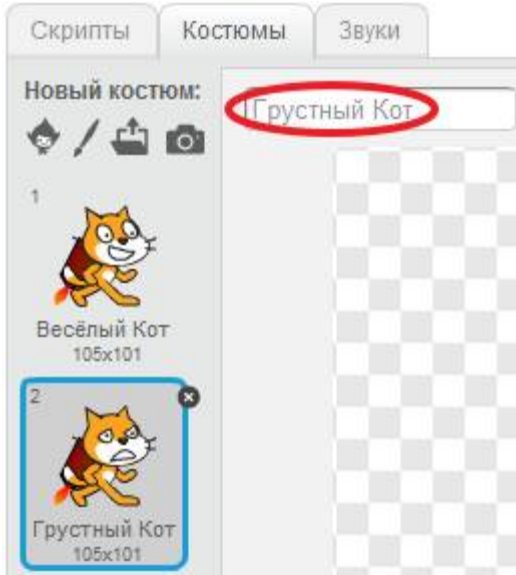
Теперь измените второй глаз.



Котёночек выглядит печальным. Осталось только переименовать костюмы. Первый костюм назовите «Весёлый Кот».



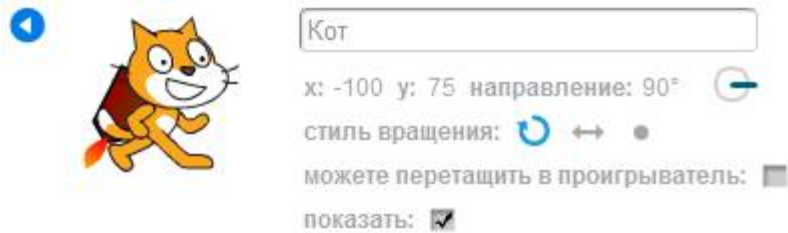
А второй костюм назовите «Грустный Кот».



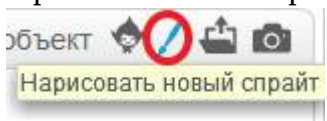
Спрайт Кот в полёту готов! Переименуйте спрайт в «Кот». Кликните на синий кружочек.



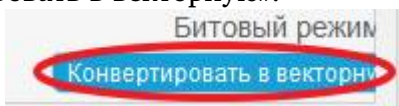
В окне свойств спрайта напишите новое имя «Кот».



Теперь надо нарисовать бревно, которое будет лететь в Кота. Нажмите на кнопку «Нарисовать новый спрайт».



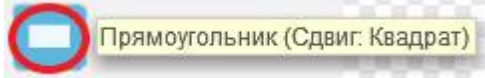
Перейдите в векторный режим рисования. Нажмите на кнопку «Конвертировать в векторную».



Сначала увеличьте масштаб — один раз нажмите на кнопку плюс.



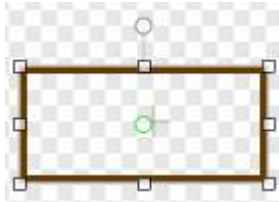
Затем выберите инструмент «Прямоугольник».



Возьмите коричневый цвет.



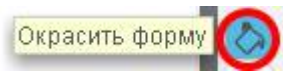
Нарисуйте прямоугольник. Обратите внимание, бревно не должно быть длинным.



Затем возьмите инструмент «Эллипс» и нарисуйте по краям два овала.



Теперь бревно надо раскрасить. Выберите инструмент «Окрасить форму».



Добавьте к коричневому цвету светло-коричневый.



Выберите тип заливки.



Закрасьте бревно.



Закрасьте один из торцов.



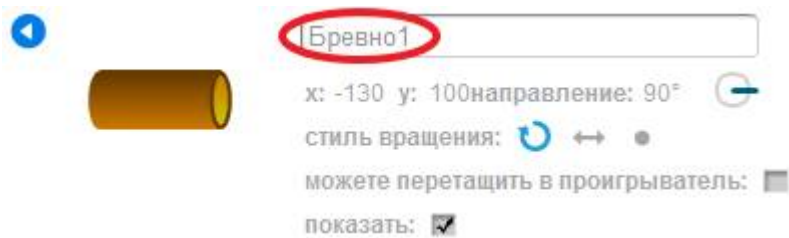
А также контурные линии.



Теперь выберите вот такие цвета и окрасьте второй торец.



Бревно готово! Осталось только переименовать его в «Бревно1». Кликните на синий кружочек, и в свойствах бревна напишите его имя.

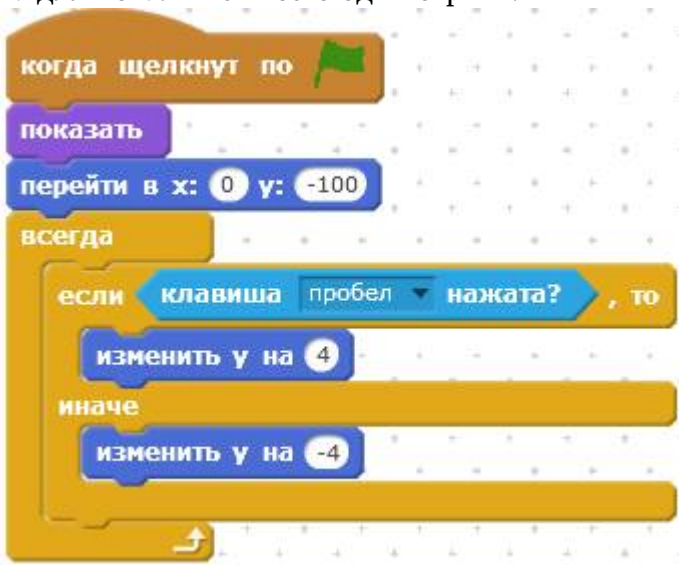


Теперь у нас есть фон, Кот и Бревно1. Можно сделать первый вариант игры.

А. ПРОСТЕЙШАЯ ИГРА, КОТ И БРЕВНО.

Игру будем делать поэтапно, постепенно усложняя скрипты.

Сначала сделаем самый простой вариант игры. Кот будет двигаться равномерно, а летать будет только одно бревно. Вот программа для Кота. В ней всего один скрипт.



В этом скрипте встречается новый блок «если иначе». Он работает очень просто, если голубое условие выполняется, ответ на вопрос «клавиша пробел нажата?» - «да», тогда координата Y у Кота измениться на четыре и Кот поднимется вверх. Если клавиша пробел не нажата, то выполниться блок «изменить Y на -4», и Кот поедет вниз. Запустите программу зелёным флажком. Нажимайте пробел — Кот летает вверх и вниз.

Теперь сделаем программу для бревна.



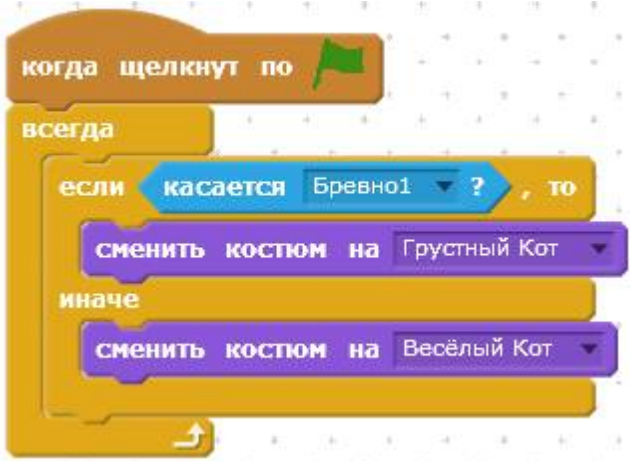
В этом скрипте нам впервые встретился зелёный блок «выдать случайное от 2 до 4». Этот блок выдаёт любое случайное число из указанного диапазона. В нашем случае блок каждый раз выдаёт либо число два, либо три, либо четыре. А из-за того, что зелёный блок лежит в блоке «ждать», следующее бревно полетит неизвестно, когда, может через две секунды, может через три, а может через четыре. Это добавляет в игру элемент неожиданности.

Ещё один новый блок — «повторять пока не», повторяет то, что спрятано у него внутри до тех пор, пока не сбудется его голубая мечта, пока бревно не «коснётся края». А внутри блока «повторять пока не» лежит блок движения бревна — блок «изменить X на -5». Как вы уже знаете, изменение X на отрицательное число означает движение влево. Таким образом, бревно будет лететь влево до тех пор, пока не «коснётся края», а как только это произойдёт, то выполнение блока прекратиться и будет выполнен блок «скрыть», и следом за ним блок «перейти в X Y» отправит бревно в начальную точку на правом краю сцены.

Снова запустите программу. Если бревно у вас не летает, то попробуйте его укоротить. Размер бревна должен быть примерно с голову Кота.



Как вы видите, бревно летает, но в Кота не попадает. Даже тогда, когда они с бревном находятся в одной точке, Кот продолжает весело улыбаться. Это надо исправить, следует добавить скрипт встречи с бревном.

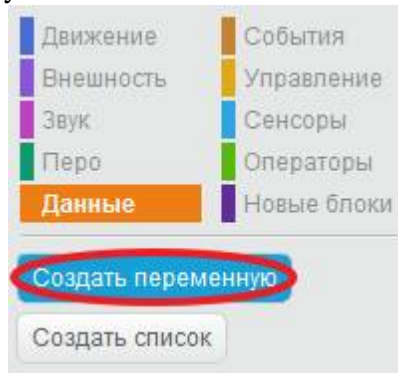


Если Кот треснется об бревно, он станет грустным, а если не треснется, то весёлым.

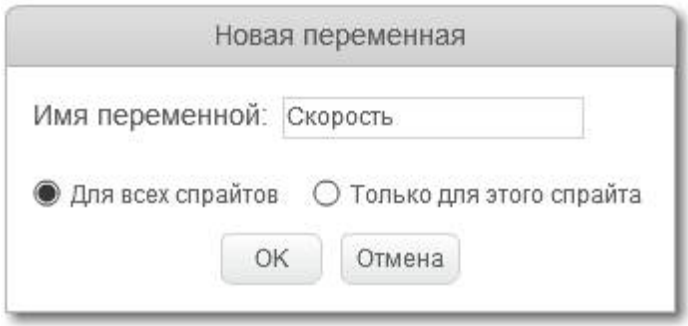
Б. УСЛОЖНЯЕМ ПРОГРАММУ. ДОБАВЛЯЕМ ФИЗИКУ, БРЁВНА И ДЕРЕВЬЯ.

У нас получилась простейшая версия программы, в которой всего одно бревно, и Кот летает как-то неестественно, плавно поднимается и опускается, как на лифте. На самом деле, когда вы летаете с реактивным ранцем, то вы летаете как на ракете, с ускорением, сначала медленно, потом всё быстрее и быстрее. Ускорение можно почувствовать в автомобиле, когда папа жмёт со свей силы на педаль газа и вас вжимает в кресло, или в самолёте, когда пилот на взлёте передвигает ручку управления двигателями от себя. С ускорением летают птички в игре Angry Birds и транспорт в игре Hill Climb Racing. Ускорение проходят в школе на уроках физики, а о компьютерных играх, в которых движение выглядит как в жизни, говорят, что они содержат физику. Давайте добавим физику и в нашу игру — полёт Кота будет больше похож на реальную жизнь, когда соседский кот надевает реактивный ранец и отправляется на крышу. Для этого нам понадобится переменная «Скорость». Чтобы создать переменную нажмите на оранжевые блоки «Данные» и на кнопку

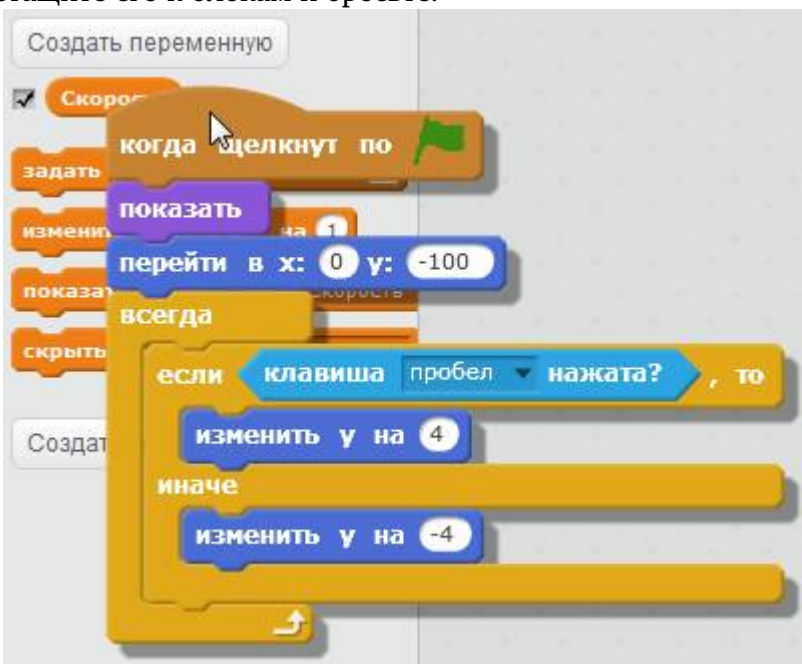
«Создать переменную».



Введите имя переменной.



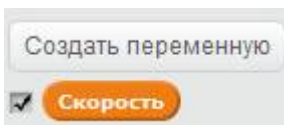
Теперь полностью удалите скрипт движения Кота. Перетащите его к блокам и бросьте.



Теперь сделайте новый скрипт движения для Котика.



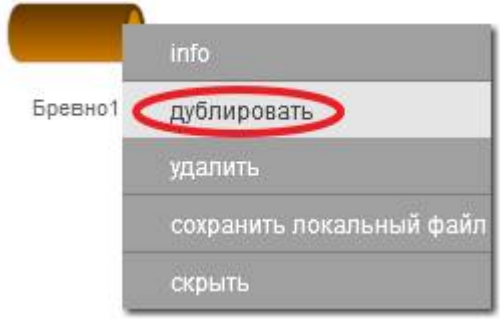
Как видите, теперь мы изменяем координату Y Кота не просто на 4 или -4, а всегда на разную величину, которая храниться в переменной «Скорость». Эта величина не постоянная. В самом начале её значение равно нулю, и, если не нажимать на «Пробел», то значение будет всё время уменьшаться, и станет меньше нуля. Y будет изменяться на отрицательную величину, а Кот будет опускаться вниз, пока не провалится за край экрана. Если Вы решите помочь ему выбраться, то нажимайте на «Пробел», значение переменной «Скорость» будет увеличиваться, и когда оно станет больше нуля, Кот полетит вверх. Если рядом с переменной стоит галочка, то вы сможете видеть её значение на экране.



Запустите игру, нажимайте на «Пробел», и внимательно понаблюдайте за значением переменной. Когда Кот летит вниз — у переменной впереди виднеется минус, когда Кот летит вверх — значение переменной больше нуля. Обратите внимание, каким плавным стал полёт Кота. Если нажать на «Пробел», то он плавно разгоняется, если отпустить, то он продолжает немного двигаться вверх по инерции. Физика работает!

Теперь давайте усложним жизнь Коту, добавим ещё два бревна. Рисовать ничего не будем. Просто скопируем наше

Бревно1 и немного изменим скрипты новых брёвен. Кликните на Бревно1 правой кнопкой и выберите «дублировать».



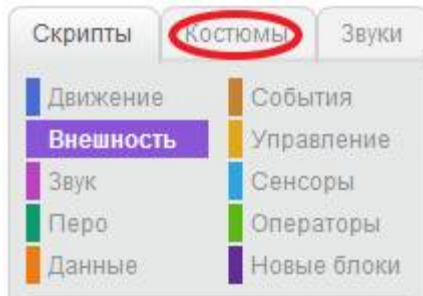
Появится спрайт второго бревна — Бревно2.



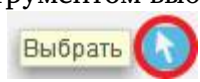
Ещё раз дублируйте первое бревно, появится Бревно3.



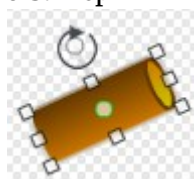
Для разнообразия можно немного изменить новые брёвна. Выберите Бревно2 и перейдите на вкладку «Костюмы».



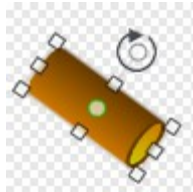
Выберите бревно инструментом выбора.



Немного поверните его за верхний кружочек.



Теперь проделайте то же самое с третьим бревном, только поверните его в другую сторону.



Вот такие брёвнышки у нас получились.



Сейчас немного изменим скрипты второго и третьего бревна, и можно будет протестировать проект. В скриптах потребуется изменить только координаты Y — то есть высоту полёта брёвен, и интервал задержки в зелёном блоке. Вот так должен выглядеть скрипт Бревна2, оно летит посередине, его координата Y всегда равна нулю. Появляется оно через каждые три, четыре, пять, или шесть секунд.

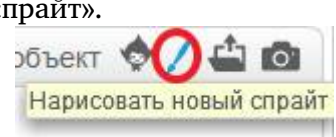


Вот скрипт Бревна3, оно летит внизу, его координата Y всегда равна -140. Это самое редкое бревно, оно летает не чаще чем раз в четыре секунды.



Запустите проект. Кот летает с большим трудом уворачиваясь от брёвен. Играть стало гораздо интереснее, но чего-то не хватает. Всё портит неподвижный фон, он не даёт поверить, что Котик летит вперёд. Для того, чтобы это исправить, нам понадобятся три тополя. Пускай они медленно плывут справа налево, это украсит игру.

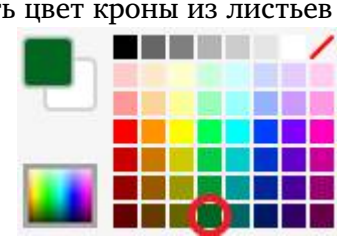
Сначала нарисуем первый тополь. Нажмите на кнопку «Нарисовать новый спрайт».



Кисть уже выбрана.



Остаётся выбрать цвет кроны из листьев и нарисовать её.





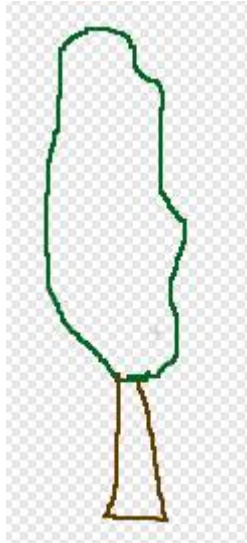
Теперь выбираем цвет ствола.



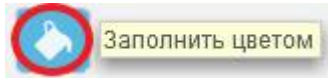
Рисуем снизу ствол.



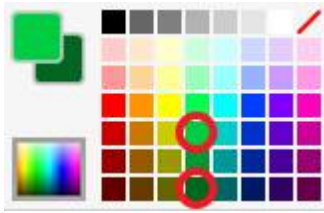
Обратите внимание, линия должна быть замкнутой, иначе мы потом не сможем закрасить крону зелёным цветом.



Выберите инструмент «Заполнить цветом».



Выберите цвета.



И способ заливки кроны.



Закрасьте крону.



Теперь выберите цвета ствола, способ заливки тот же.



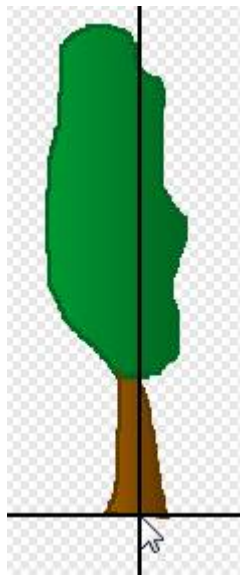
Закрасьте ствол.



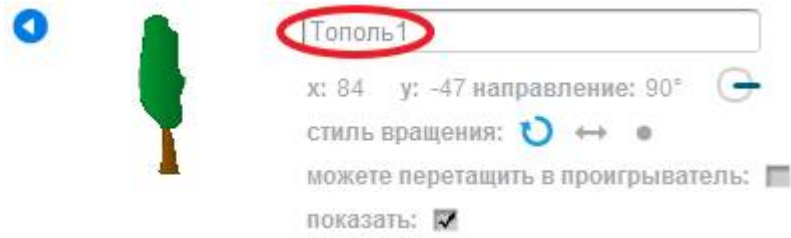
Самое главное, не забыть установить центр костюма в самом низу дерева. Кликните на кнопку «Установить центр костюма».



Кликните внизу ствола.



Тополь готов! Переименуйте его.



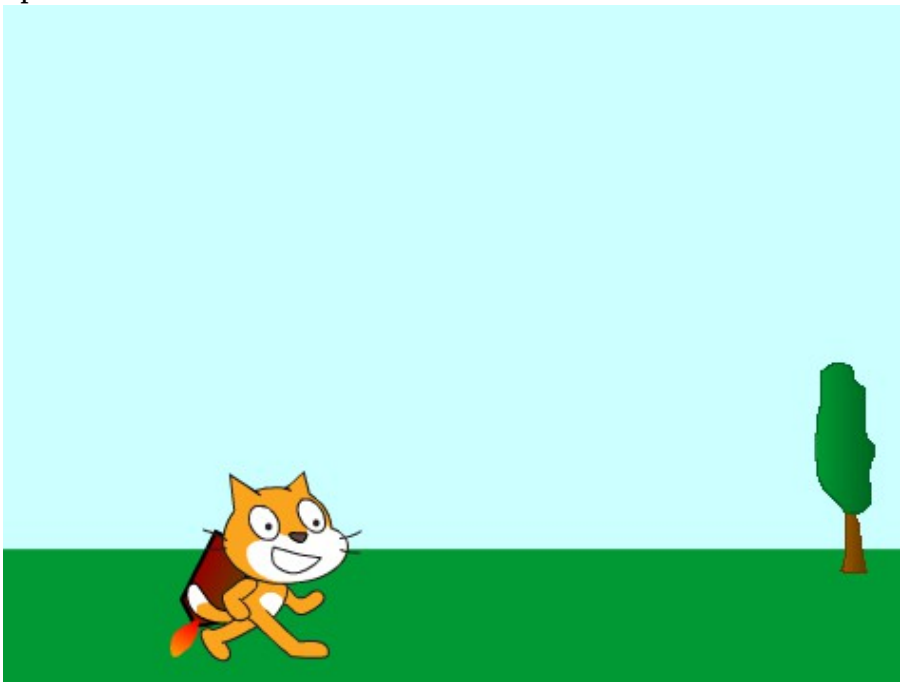
Сделайте Тополю1 такую программу.



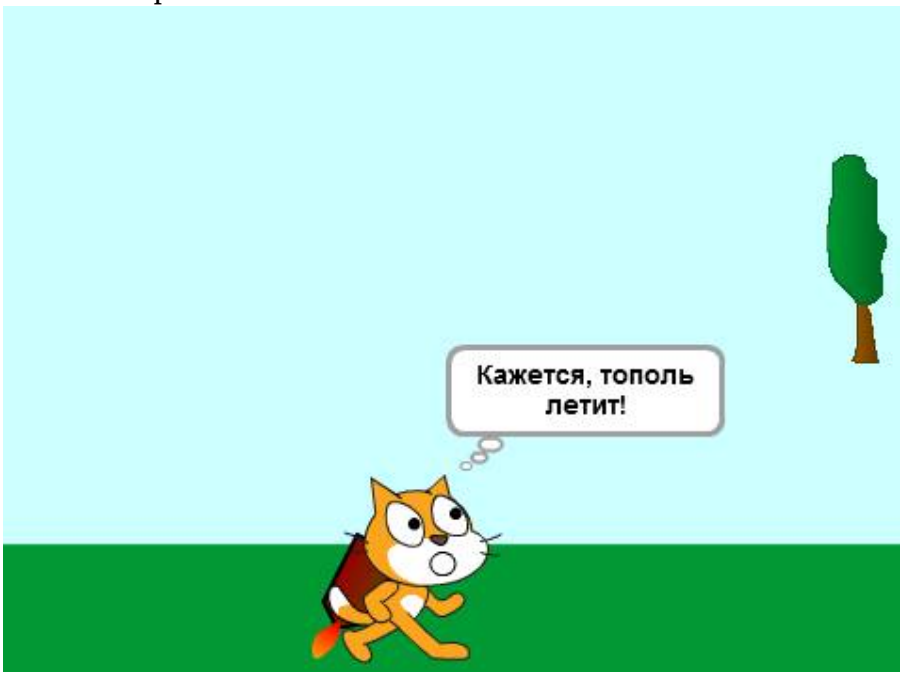
Программа очень похожа на программу бревна. Главное отличие в блоке «изменить размер на 0.3». Этот блок постоянно увеличивает дерево во время движения влево, и нам кажется, что тополь приближается. Запустите проект, посмотрите, как двигается дерево. Если дерево не появляется, или начинает движение падая с неба, то скрипту тополя потребуется отладка. Надо будет подобрать подходящие значения для координат начальной точки X = 190 и Y = -90, для начального размера 40%, и для скорости изменения размера 0.3.

Координаты X и Y вам надо будет подобрать самостоятельно так, чтобы в момент начала движения ствол дерева касался травы и не касался края сцены. Если ваше дерево шире моего, то уменьшайте X, сделайте его 170 или 150. Вот правильное начало

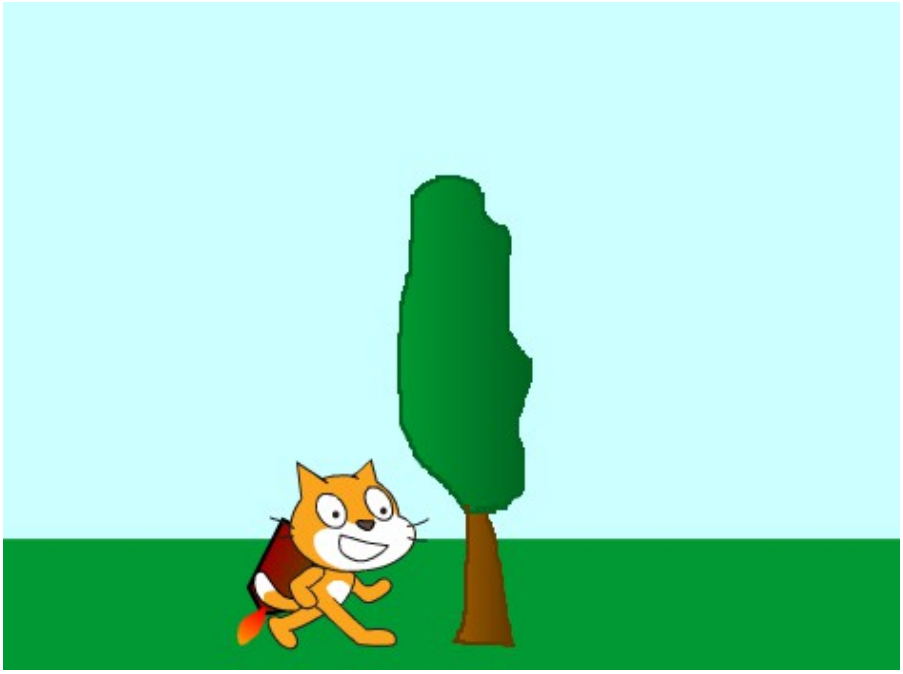
игры.



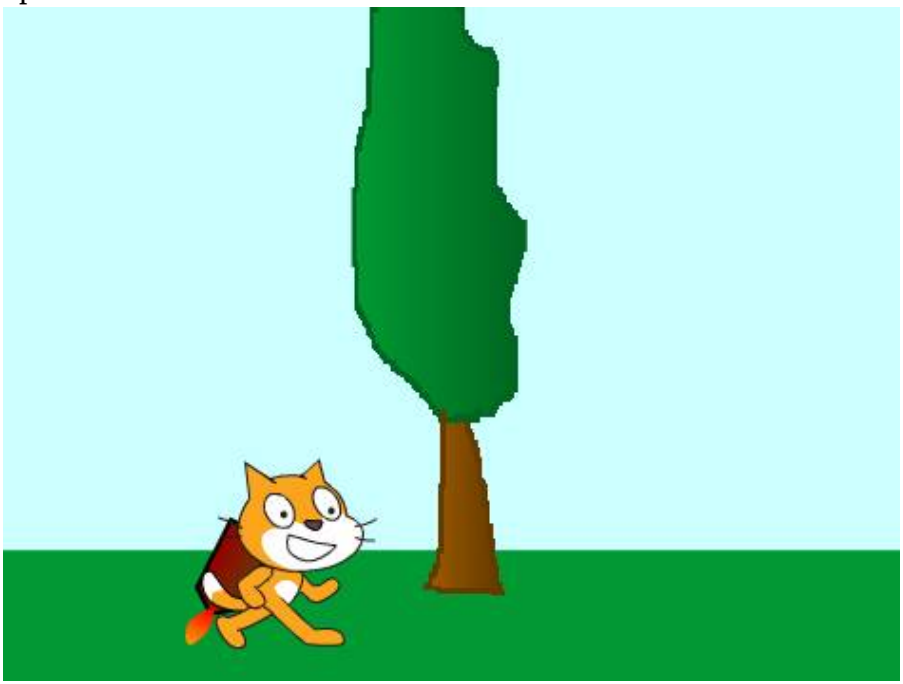
Вот неправильное начало.



Начальный размер в 40% также подбирается на глаз. Так как я нарисовал очень большое дерево, то понадобилось его уменьшить до сорока процентов. Вот так выглядит мой тополь при размере 100% по сравнению с Котом.

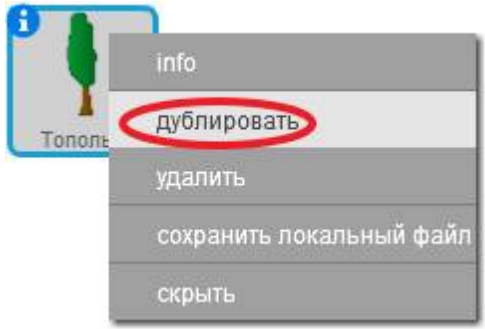


Если вы нарисуете не такое большое дерево как у меня, от и процент у вас будет отличаться в большую сторону. Когда работа скрипта первого дерева будет отлажена, дерево должно плыть до левого края экрана и исчезать. Если дерево исчезает немного раньше, то, наверное, оно коснулось не левого края экрана, а верхнего или нижнего.



В этом случае вы можете изменить значение в блоке «изменить размер на 0.3». Напишите там 0.25 или 0.2.

Когда отладка скрипта Тополя1 закончена, то его смело можно дублировать два раза подряд.



Теперь у нас три тополя.



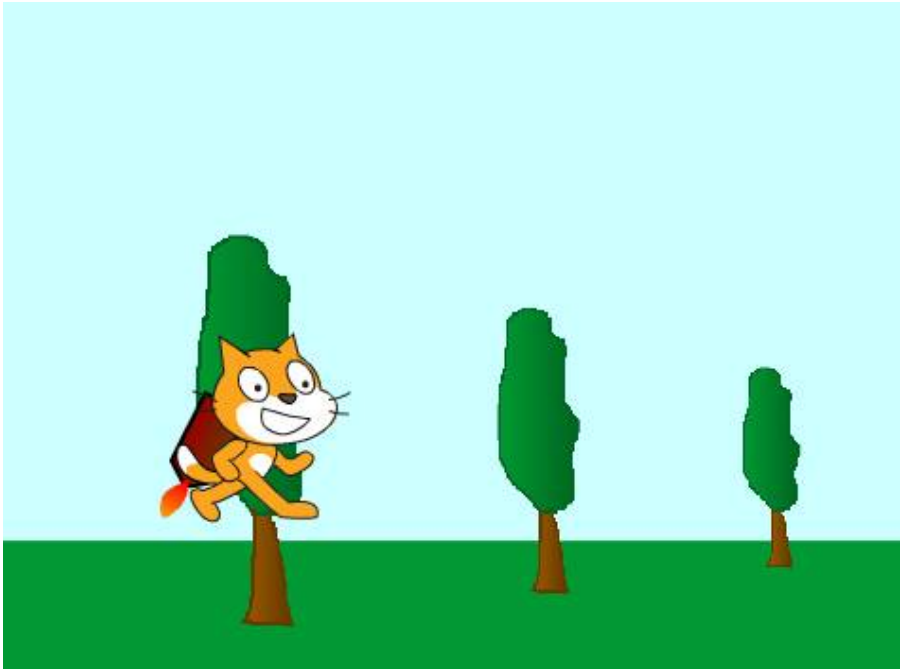
Скрипты Тополя2 и Тополя3 отличаются от первого только блоком «ждать». Вот скрипт Тополя2.



Вот скрипт Тополя3.



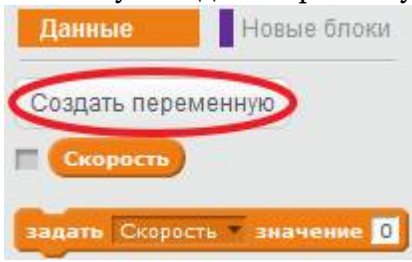
Запускайте проект. Если первое дерево двигалось правильно, то и теперь все три тополя будут двигаться также. Отладка больше не потребуется.



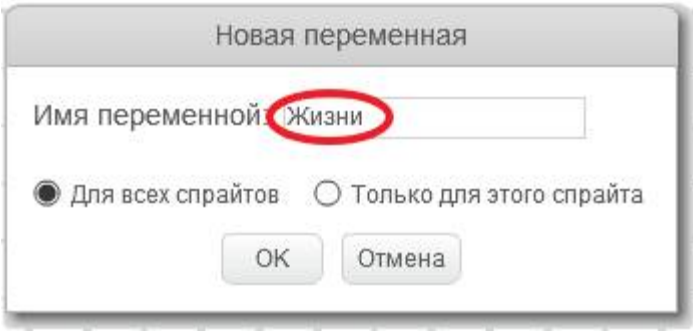
В. ФИНАЛЬНАЯ ДОРАБОТКА ИГРЫ.

Добавим Коту Жизни, которые будут убавляться при столкновении с брёвнами, а также сделаем так, чтобы он не улетал за край сцены.

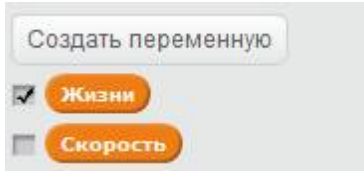
Сначала добавим переменную «Жизни». Выберите оранжевые блоки и нажмите на кнопку «Создать переменную».



Введите имя переменной.



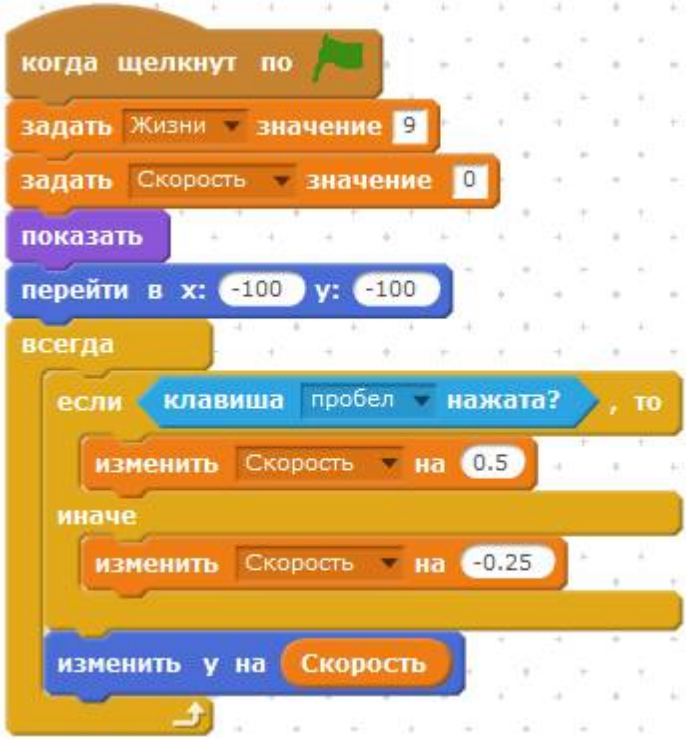
Появится новая переменная.



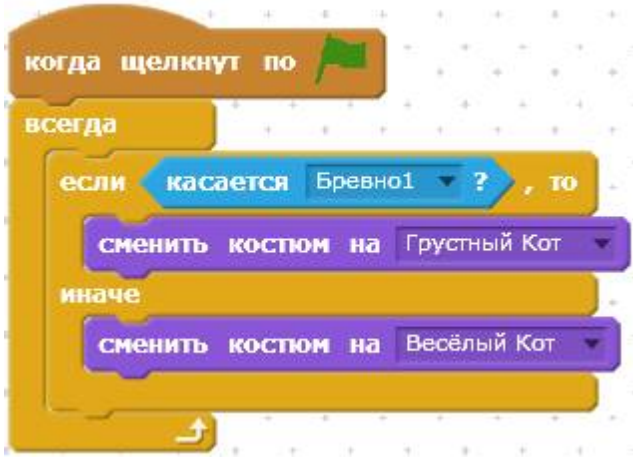
Установите около неё галочку, чтобы переменная отображалась на экране.



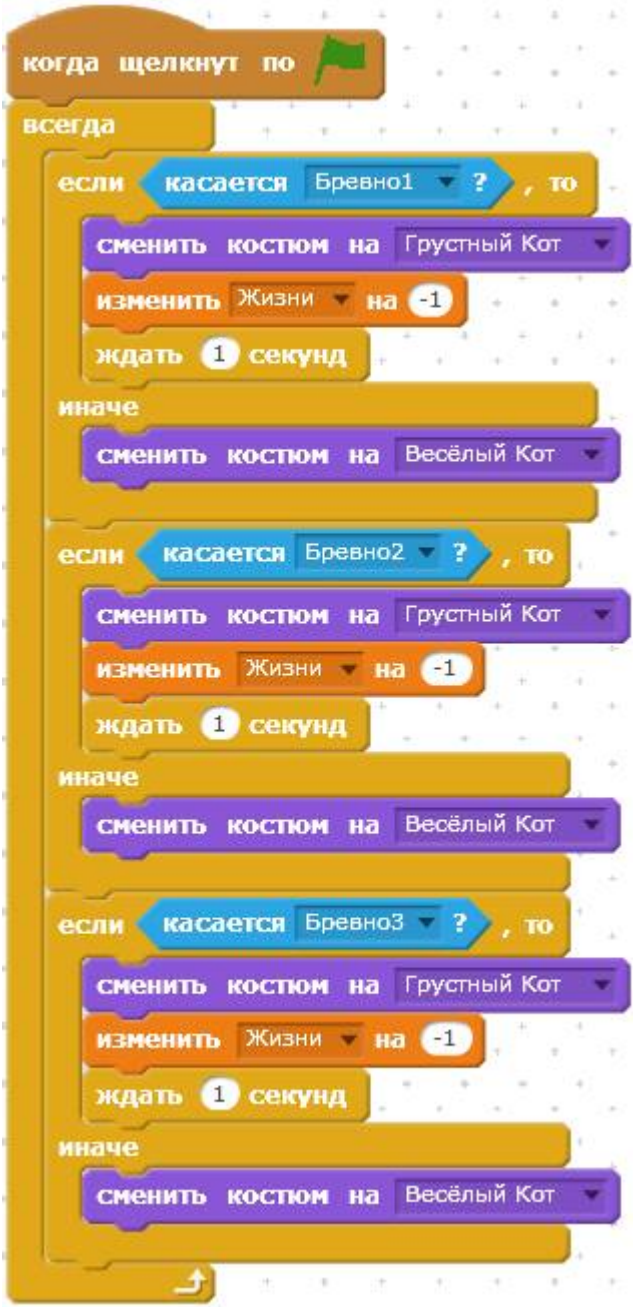
Доработайте первый скрипт Кота. Добавьте туда блок «задать Жизни значение 9». Теперь у кота будет всего 9 жизней.



Теперь доработайте скрипт столкновения Кота с препятствием. Сейчас он выглядит вот так.



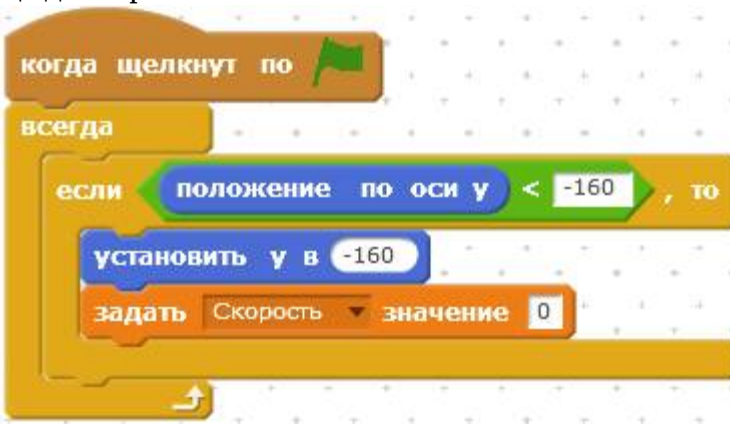
А должен выглядеть вот так.



Давайте посмотрим, как работает этот скрипт. В блок «всегда» вложены три одинаковых блока «если иначе», каждый из которых реагирует на своё бревно. При столкновении с бревном Котик меняет костюм на грустный, количество его жизней уменьшается на единицу, а как только бревно пролетит дальше, и перестанет касаться Кота, он снова станет весёлым.

Для того, чтобы Котик не улетал за край сцены, добавьте

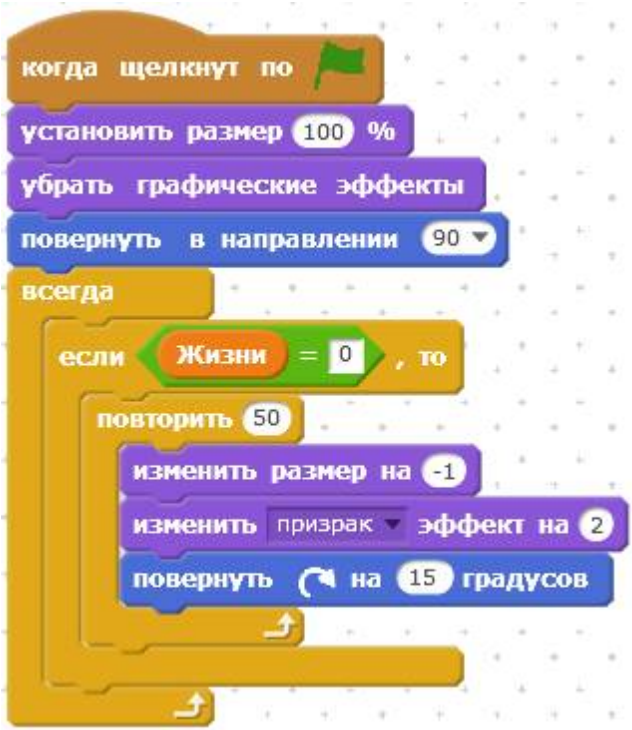
Коту ещё два скрипта.





Первый скрипт работает следующим образом, если Кот долетел до самого низа сцены, и его координата Y стала меньше чем -160, то дальше координата Y изменяться не будет. Y будет установлен в -160, Кот застынет в нижней точке. Второй скрипт работает похожим образом. Если Кот долетел до самого верха сцены, и его координата Y стала больше чем 160, то дальше координата Y изменяться не будет. Y будет установлен в 160, Кот застынет в верхней точке. Значение скорости в обоих случаях будет равно нулю.

Осталось сделать только последний скрипт — скрипт проигрыша. Если у Кота останется ноль жизней, то он должен вращаясь уменьшиться и исчезнуть.



Вот теперь игра готова. Постарайтесь продержаться как можно дольше.

Вопрос: как ускорить полет брёвен?



Задание: сделайте, чтоб все три бревна вращались в полёте.

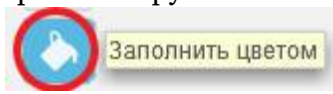
ГЛАВА 15. ИГРА ПЛАТФОРМЕР.

Давайте сделаем простейший платформер. Кот бежит домой через три экрана. Если Кот упадет в красную лаву, то конец игры.

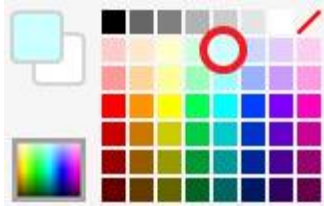
Игра будет выглядеть вот так.



Сначала нарисуем фон. Выберите сцену и залейте весь фон голубым цветом. Выберите инструмент «Заполнить цветом».



Возьмите голубой цвет.

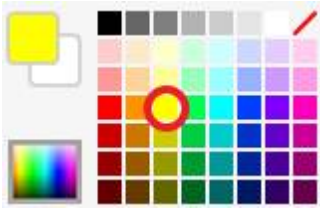


Залейте фон голубым.

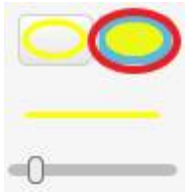
Потом нарисуйте солнце инструментом «Эллипс».



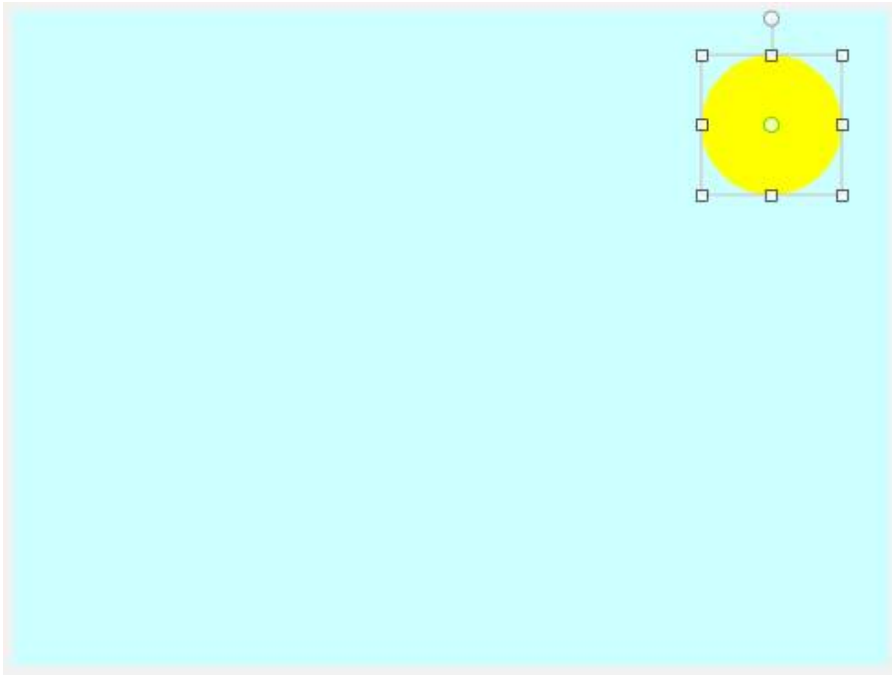
Цвет, конечно, жёлтый.



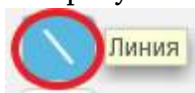
Выберите закрашенный овал.



Удерживая клавишу Shift, нарисуйте ровный круг.



Инструментом «Линия» нарисуйте солнечные лучи.

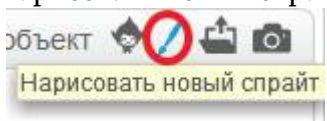


Фон готов.

Земля и лава будут нарисованы в виде отдельных спрайтов.

Кот всегда будет находиться в центре экрана, и двигаться только вверх и вниз. Спрайты с землёй и лавой перемещаясь справа налево будут создавать эффект движения. Игроку будет казаться, что это не земля двигается влево, а что Кот идёт вправо.

Нарисуйте 4 спрайта с землей и лавой. Постарайтесь участки лавы делать не очень широкими. Нарисуйте первый спрайт. Нажмите на кнопку «Нарисовать новый спрайт».



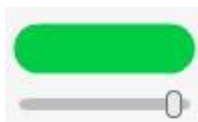
Выберите инструмент «Кисть».



Возьмите зелёный цвет.



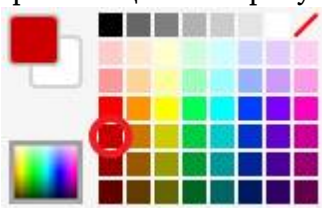
Толщину кисти сделайте почти самую большую, широкой кистью удобней рисовать.



Нарисуйте траву.



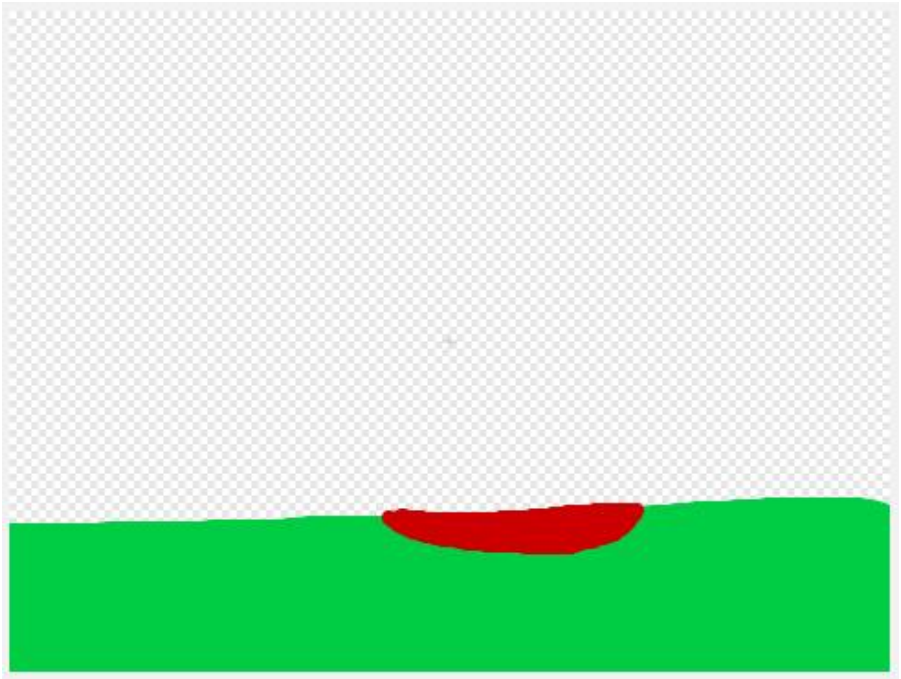
Теперь выберите красный цвет и нарисуйте лаву.



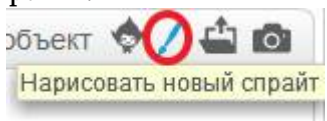
Толщину кисти лучше немного уменьшить.



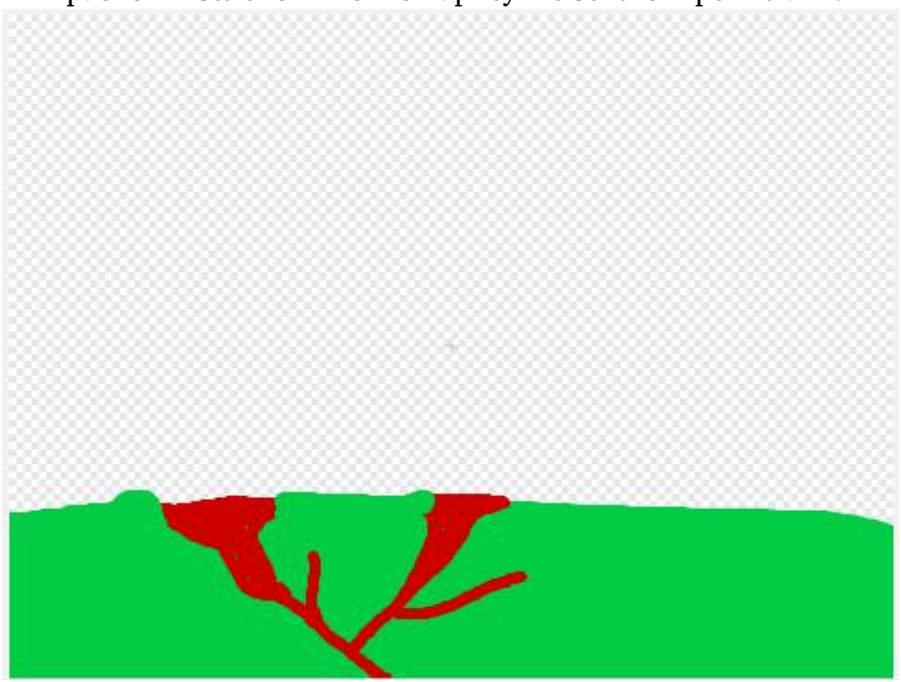
Нарисуйте не очень большое озеро лавы.



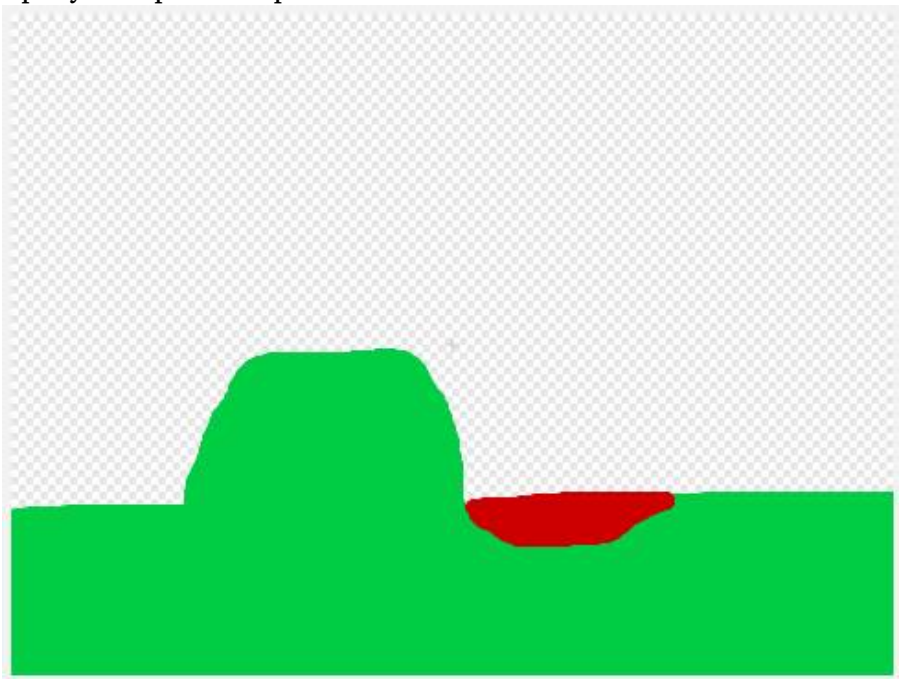
Затем нарисуйте второй спрайт. Нажмите на кнопку «Нарисовать новый спрайт».



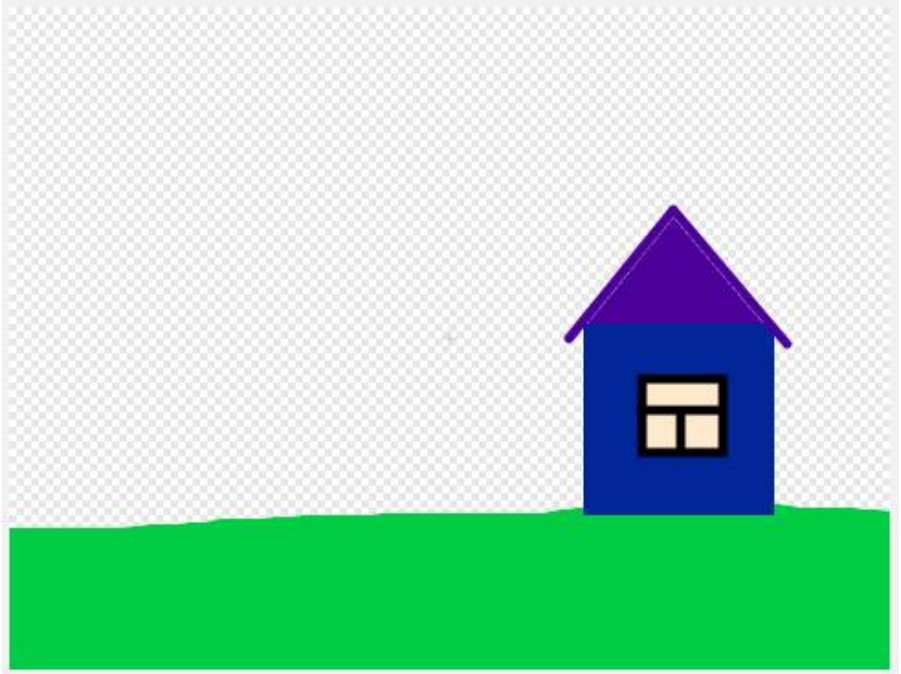
Красной и зелёной кистью нарисуйте землю и реки лавы.



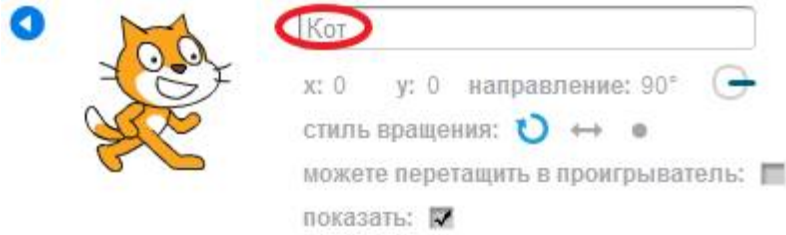
Снова нажмите на кнопку «Нарисовать новый спрайт» и нарисуйте третий спрайт.



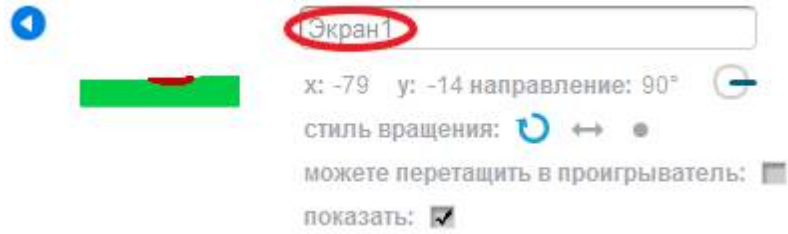
А теперь нарисуйте четвёртый спрайт. Лаву на нём рисовать не надо, нарисуйте домик Кота с помощью инструментов «Прямоугольник, «Линия», «Заполнить цветом». Цвет домика должен быть синим.



Переименуйте все имеющиеся спрайты. Кота назовите Котом.



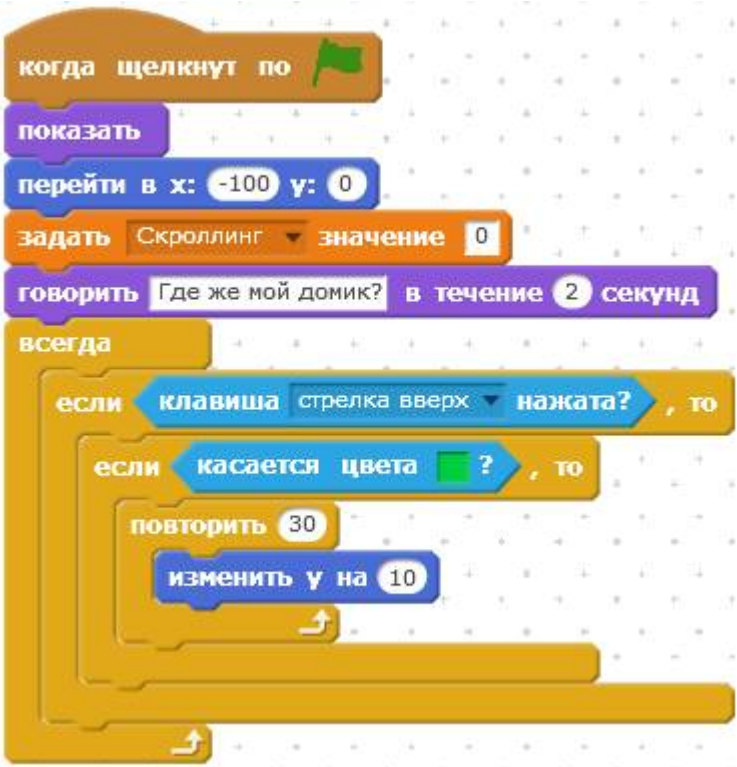
Первую картинку назовите Экран1.



Следующим спрайтам дайте имена Экран2, Экран3 и Экран4.

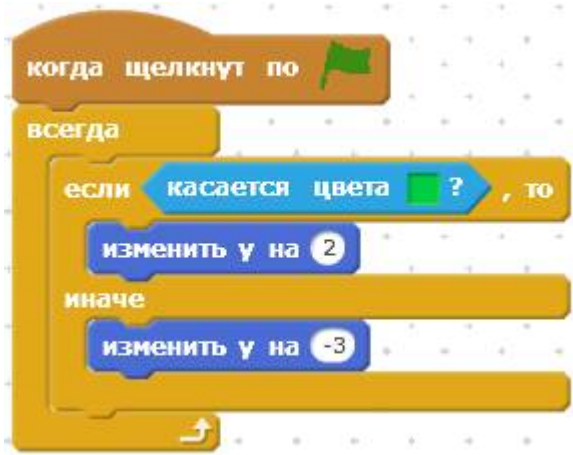


Начинаем программировать. Сделайте программу Кота. В ней будет шесть скриптов. В первом скрипте описано начало программы. Кот появится, перейдёт в исходную точку, и спросит где его домик. Заодно обнулится переменная «Скроллинг».


когда щелкнут по
показать
перейти в x: -100 y: 0
задать Скроллинг значение 0
говорить Где же мой домик? в течение 2 секунд
всегда
если клавиша стрелка вверх нажата? , то
если касается цвета ? , то
повторить 30
изменить y на 10

В блоке «всегда» запрограммирован прыжок Кота. Для того, чтобы Кот подпрыгнул, должно выполниться два условия. Должна быть нажата стрелка вверх и Кот должен касаться зелёного цвета. То есть, если стрелка вверх будет нажата, когда Кот стоит на земле и касается зелёного цвета, то он подпрыгнет вверх, изменяя координату Y тридцать раз по десять, то есть на триста пикселей вверх. А если стрелку вверх нажать во время прыжка, то есть когда кот не касается зелёного, то он не подпрыгнет. Из лавы выпрыгнуть тоже не получится, так как она красного цвета.

Второй скрипт Кота отвечает за то, чтобы Кот мог ходить по земле и не проваливаться.



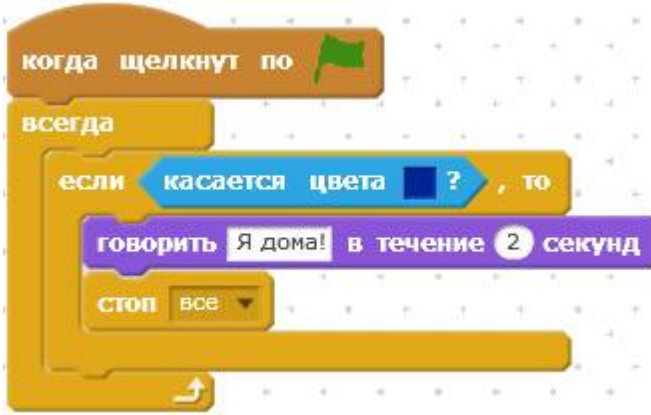
Если Кот касается зелёного цвета, то он понемногу поднимается вверх, изменяя Y на два пикселя. Как только Кот окажется на поверхности земли, то он немного подпрыгнет и перестанет касаться зелёного цвета. Тут же сработает блок «изменить Y на -3», и он опять коснётся земли. В результате Кот будет немного дрожать, стоя на земле. Это не страшно, играть всё равно интересно.

Третий скрипт Кота отвечает за касание лавы.



Если Кот коснётся красного цвета, то скажет печальную фразу, и скроется. Работа программы будет остановлена.

Четвёртый скрипт отвечает за возвращение Кота домой. Если Кот дойдёт до дома, и коснётся синего цвета, то он скажет весёлую фразу и наступит конец игры.

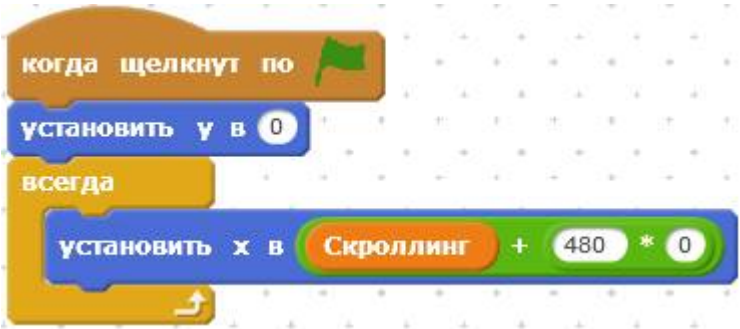


Последние два скрипта отвечают за движение Кота влево и вправо.



Кот управляется стрелками, но не как обычно, изменяя координату X, а изменяя значение переменной. Ведь на самом деле в этом проекте Кот никуда не двигается. Двигаются только спрайты Экран1, Экран2, Экран3 и Экран4. Если мы нажимаем стрелку влево, то Экраны будут смещаться вправо. Из-за этого будет казаться, что Кот идёт влево.

Теперь сделайте программу для Экрана1. В ней всего один скрипт



Давайте внимательно посмотрим, как работает этот скрипт. В самом начале работы программы переменная «Скроллинг» равна нулю. 480 умноженное на ноль, тоже равно нулю. Ноль плюс ноль тоже равно нулю. Значит в начале игры Экран1 будет стоять в точке (X = 0; Y = 0) которая соответствует центру сцены.



Совет. Если вытащить формулу и дважды кликнуть на неё, то вверху отобразиться её значение.



У Экрана2 программа немного другая.



Здесь 480 умножается на один. Значит в начале работы программы Экран2 будет находиться в точке (X = 480; Y = 0), а это справа за границей экрана, поэтому Экран2 совсем не видно.



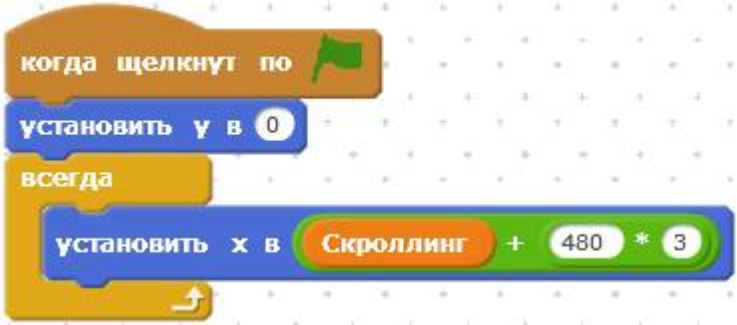
У Экрана3 программа очень похожа на программу Экрана1 и Экрана2.



Отличается только числом, на которое умножается 480. В начале игры Экран3 окажется в точке (X = 960; Y = 0), а это справа за границей экрана, еще правее чем Экран2.



Какая программа будет у Экрана4 вы уже, наверное, догадались.



Таким образом, все экраны расположились в ряд.



Запустите игру. Попытайтесь пройти Котом до домика. Не касайтесь лавы!

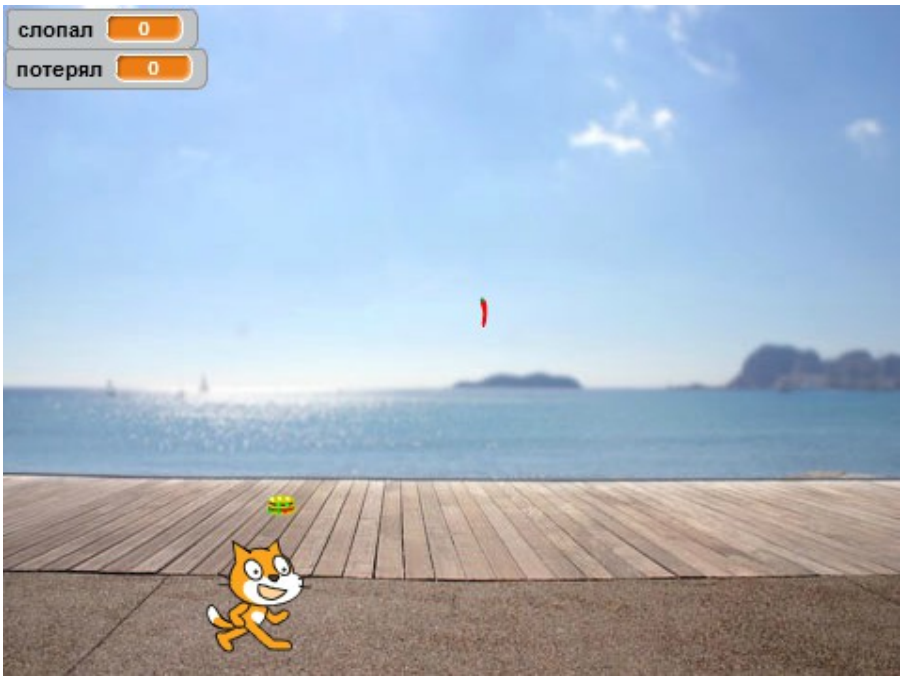
Вопросы. Как ускорить движение Кота.



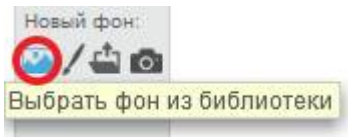
Задание. Подрисуйте ещё два Экрана так, чтобы домик Кота находился на Экране6.

ГЛАВА 16. ИГРА «ЛОВИ ВКУСНЯШКИ».

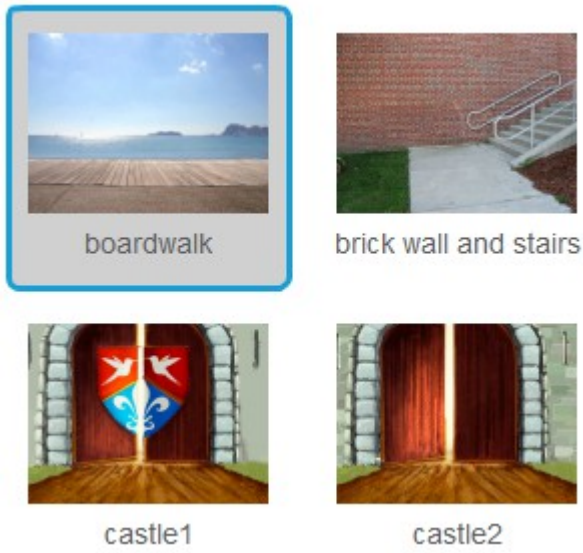
В этой игре Кот, управляемый мышкой, бегает влево и вправо. С неба падают вкусняшки, которые надо ловить ртом. Если Кот поймал вкусняшку, то переменная «слопал» увеличится на единицу. Если вкусняшка упала на землю, или Кот наелся перца, то на единицу увеличится переменная «потерял». Цель игры — набрать как можно больше очков «слопал». Если переменная «потерял» достигла пяти, то наступит конец игры. В этой игре мы будем использовать переменные, блоки случайных чисел, научимся составлять сложные условия. Игра будет выглядеть вот так.



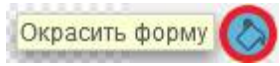
Для начала надо выбрать из библиотеки фонов какой-нибудь симпатичный фон. Нажмите на кнопку «Выбрать фон из библиотеки».



Мне понравился вот такой фон. Выберите его и нажмите внизу «Ок».



Теперь Котику надо закрасить рот светло-оранжевым. Выберите его, перейдите на вкладку Костюмы, и кликните на инструмент заливки.



Выберите цвет.



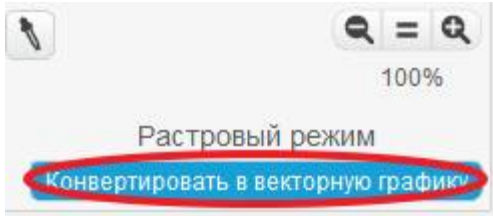
Закрасьте Котику рот.



Теперь надо нарисовать вкусняшки, которые будут падать с неба. Все они будут разными костюмами одного спрайта. Нажмите на кнопку «Нарисовать новый спрайт».



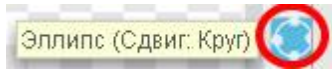
Перейдите в векторный режим рисования.



Сначала нарисуем пирожное. Так как оно будет очень маленькое, то увеличьте масштаб до 1600%.



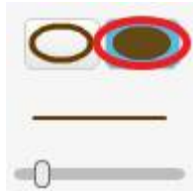
Выберите инструмент «Эллипс».



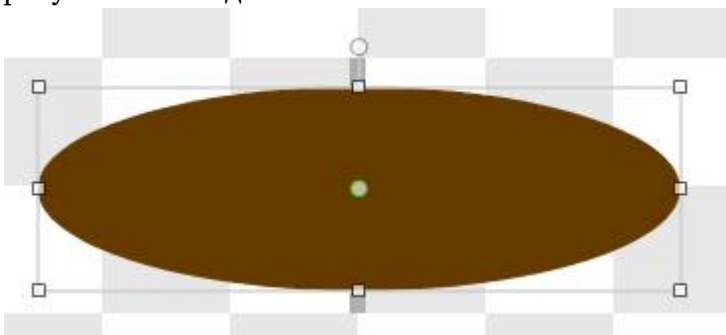
Возьмите шоколадный цвет.



Выберите закрашенный вид овала.



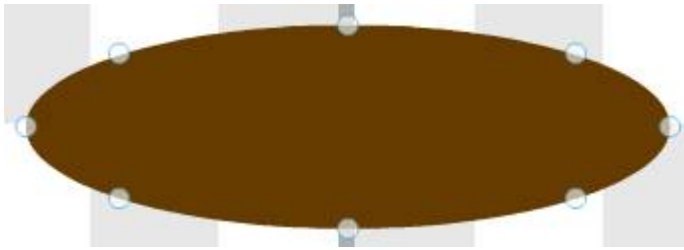
Нарисуйте шоколадный овал.



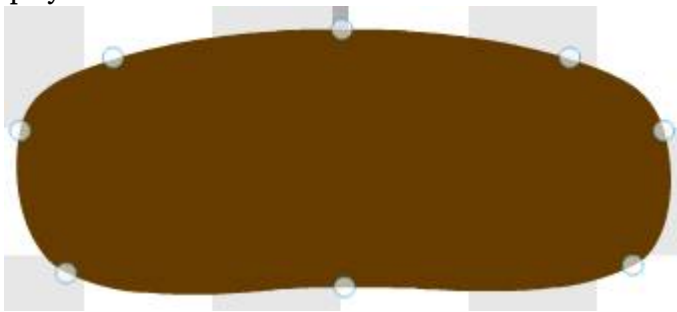
Выберите инструмент «Изменение формы».



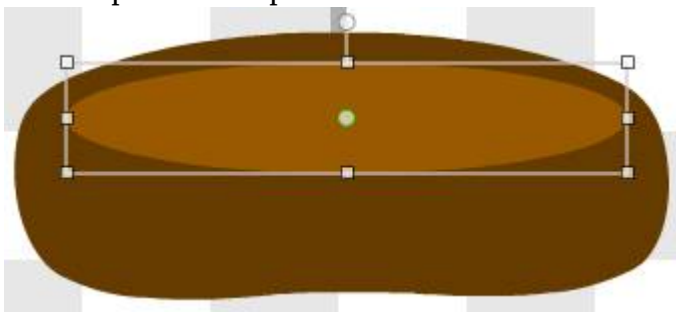
Кликните на овал — он покроется точками привязки.



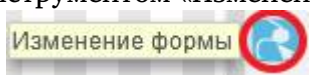
Перетаскивая три нижние точки привязки придайте овалу такую форму.



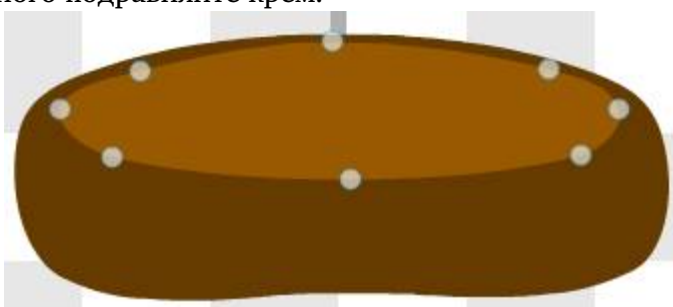
Теперь снова выберите инструмент эллипс, и нарисуйте сверху светло-коричневый крем.



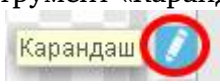
Выделите крем инструментом «Изменение формы».



Немного подравняйте крем.

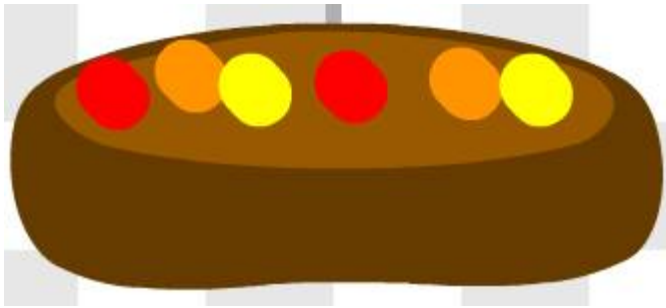


Теперь выберите инструмент «Карандаш».

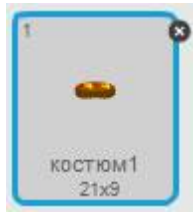


Выбирайте разные цвета и нарисуйте сверху сладкую

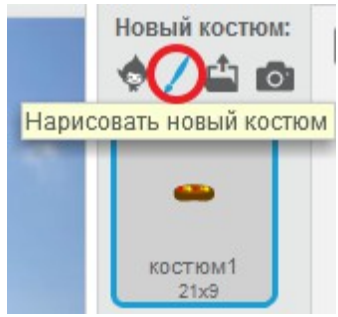
посыпку.



Первый костюм готов.



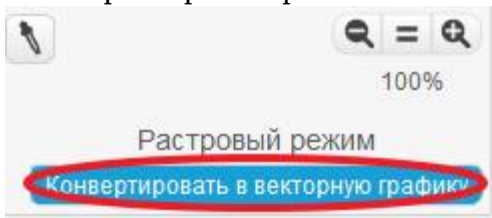
Давайте нарисуем второй костюм. Нажмите на кнопку «Нарисовать новый костюм».



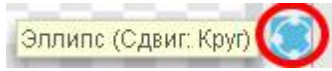
Нарисуйте гамбургер. Так как он тоже будет очень маленьким, то увеличьте масштаб до 1600%.



Перейдите в векторный режим рисования.



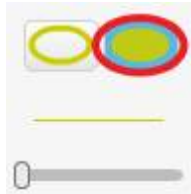
Выберите инструмент «Эллипс».



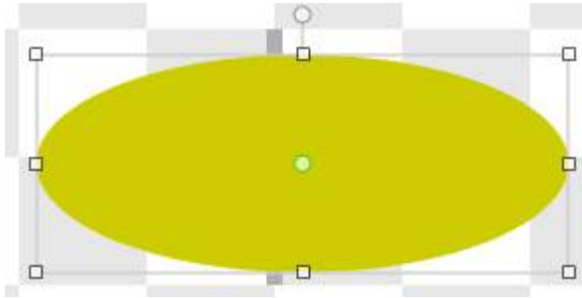
Возьмите вот такой цвет.



Кликните закрашенный овал.



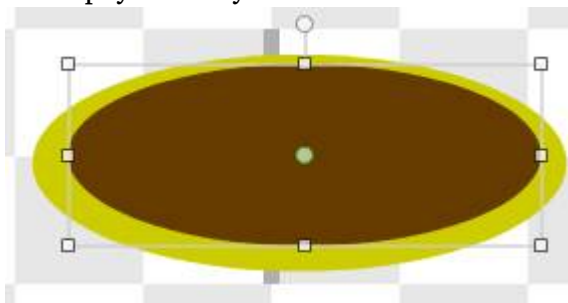
Нарисуйте овальчик.



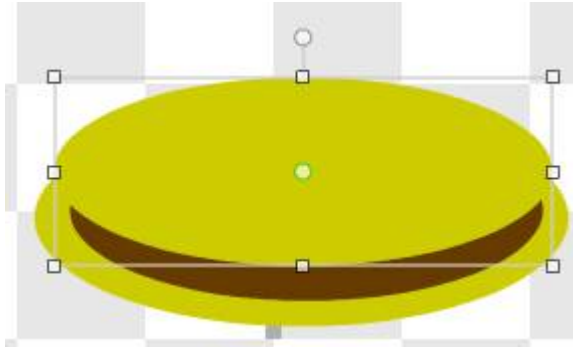
Затем возьмите коричневый цвет.



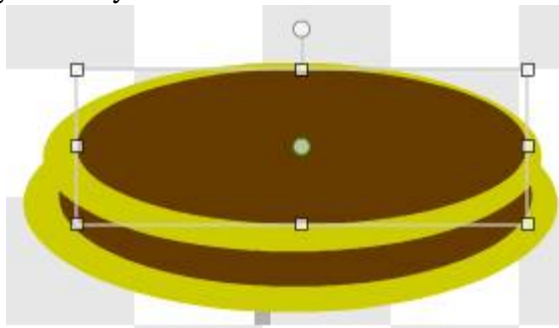
Нарисуйте сверху котлету.



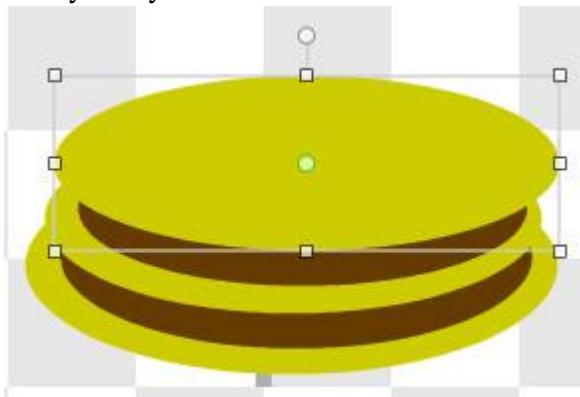
Затем снова нарисуйте булочку.



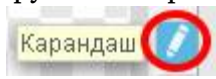
Ещё одну котлету.



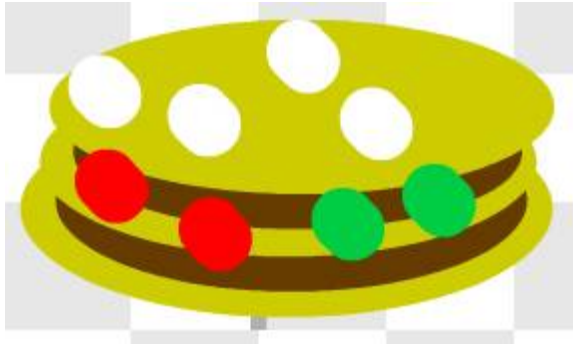
И верхнюю булочку.



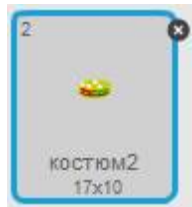
Теперь выберите инструмент «Карандаш».



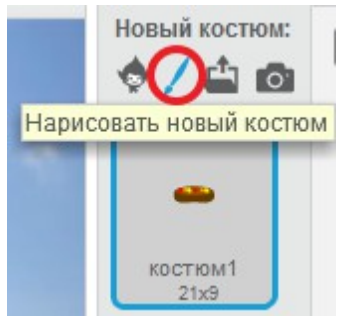
Меняя цвета и толщину карандаша, нарисуйте листики салата, помидорчики, и кунжут.



Второй костюм готов.



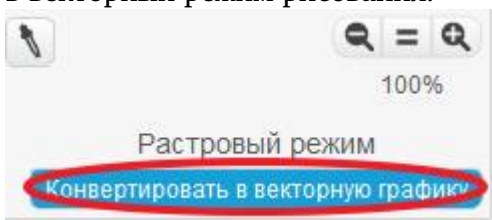
Теперь нарисуем помидорчик. Нажмите на кнопку «Нарисовать новый костюм».



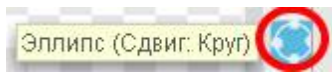
Увеличьте масштаб до 1600%.



Перейдите в векторный режим рисования.



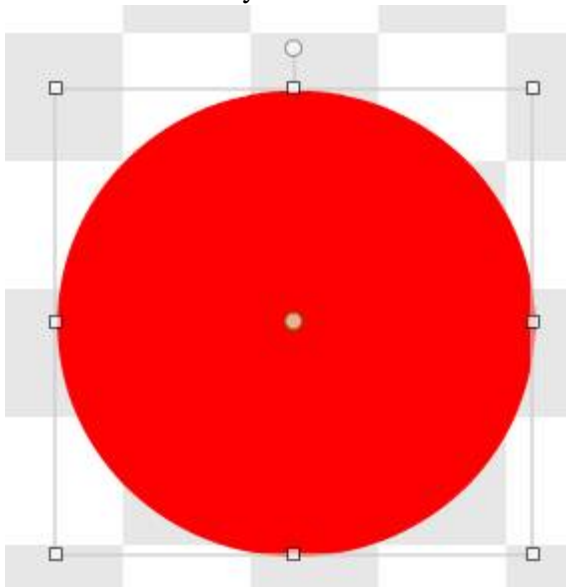
Выберите инструмент «Эллипс».



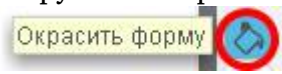
Возьмите красный цвет.



Нарисуйте ровный круг. Для этого при рисовании удерживайте нажатой клавишу Shift.



Теперь выберите инструмент «Окрасить форму».



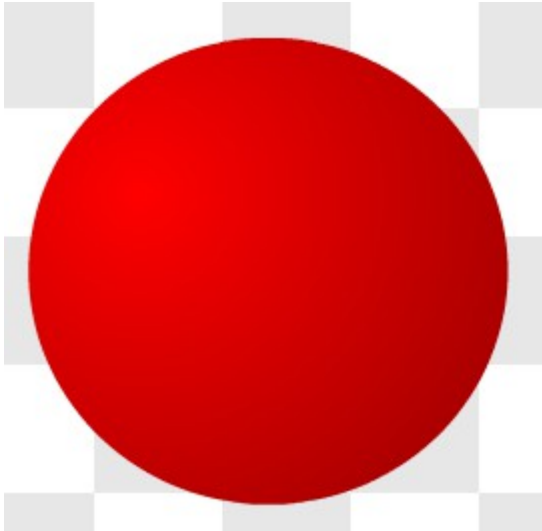
Выберите такие цвета.



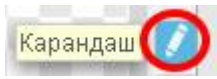
И вот такой способ заливки.



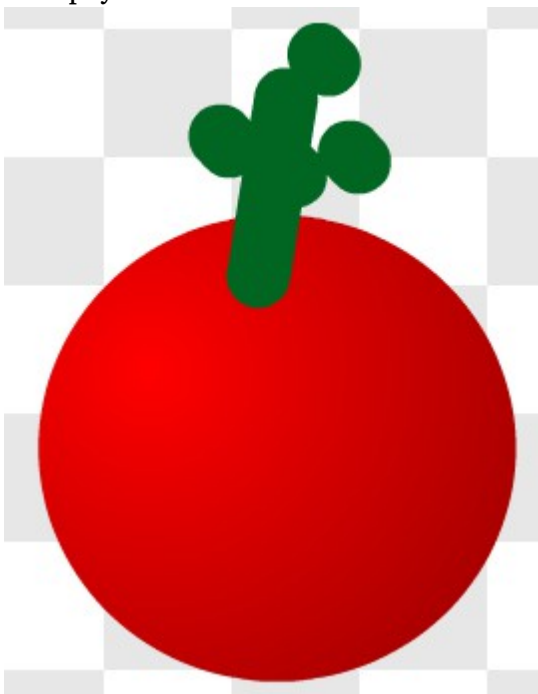
Закрасьте помидор.



Осталось только подрисовать зелёные листики с помощью инструмента «Карандаш».



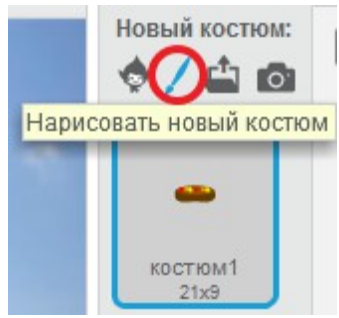
Нарисуйте сверху листики.



Получился вот такой помидорчик.



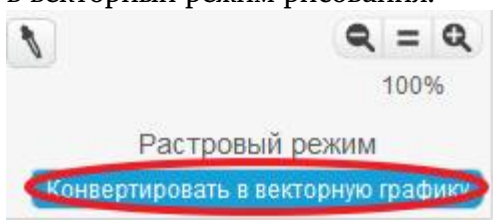
А теперь нарисуем перчик чили. Он будет очень жгучий, поэтому котёнку его лучше не пробовать. Нажмите на кнопку «Нарисовать новый костюм».



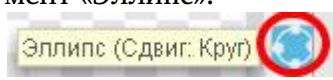
Увеличьте масштаб до 1600%.



Перейдите в векторный режим рисования.



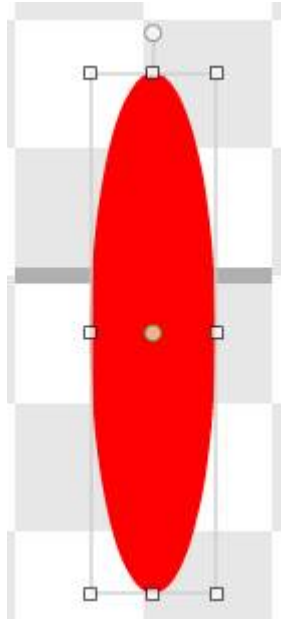
Выберите инструмент «Эллипс».



Возьмите красный цвет.



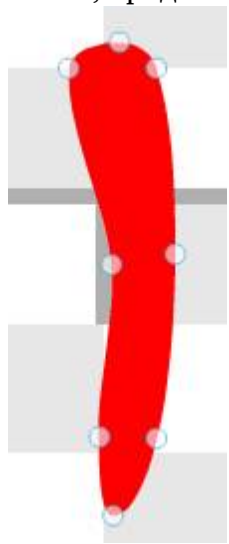
Нарисуйте вытянутый овал.



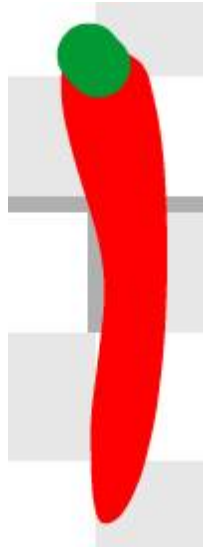
Теперь возьмите инструмент «Изменение формы».



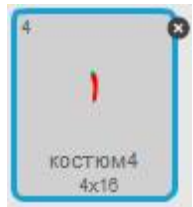
Перемещая точки привязки, придайте овалу форму перца.



Подрисуйте листик зелёного цвета.



Перчик тоже готов!



Начинаем программировать!

Сделайте программу для Кота. В ней будет два скрипта.

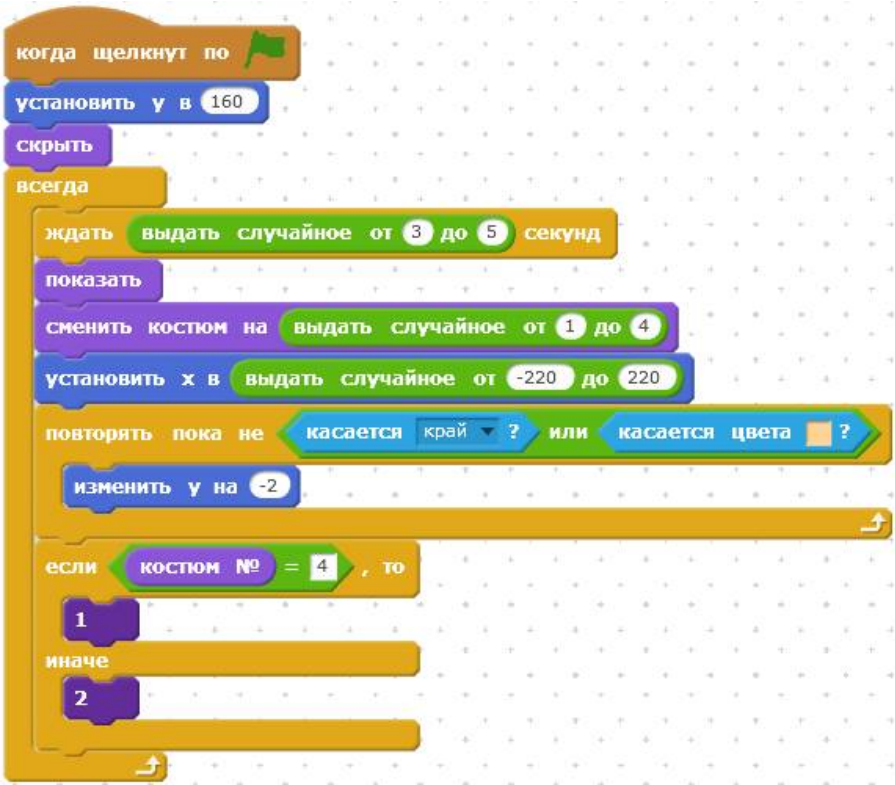


Первый скрипт очень простой. Он заставляет следовать Кота за курсором мышки влево и вправо. Второй скрипт немного сложнее.

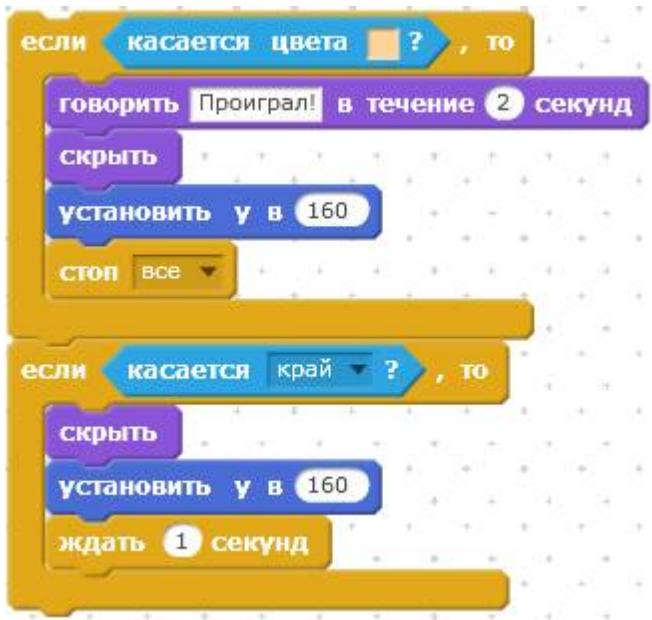


Сначала в этом скрипте мы обнуляем две переменных. Переменная «слопал» считает количество вкусняшек, которые попали Коту в рот. Переменная «потерял» считает количество вкусняшек, которые упали мимо.

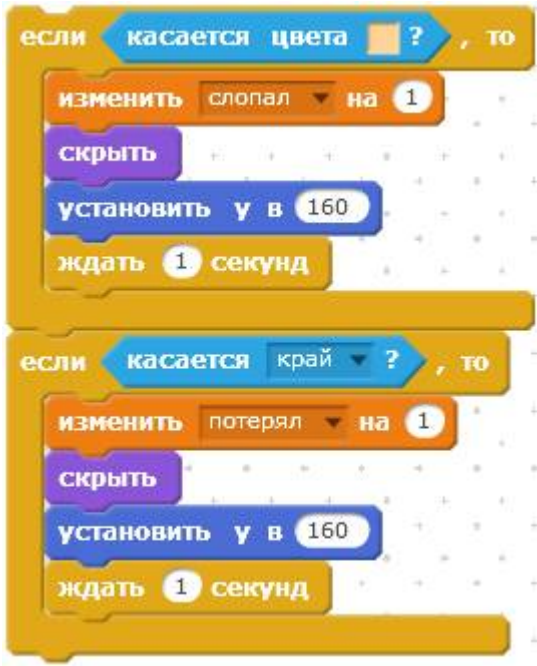
Теперь сделайте скрипт для вкусняшки. Он очень большой. Для того, чтобы он поместился на экран, я немного разобрал его на части. Сначала сделайте первую часть скрипта.


когда щелкнут по
установить y в 160
скрыть
всегда
ждать выдать случайное от 3 до 5 секунд
показать
сменить костюм на выдать случайное от 1 до 4
установить x в выдать случайное от -220 до 220
повторять пока не касается край ? или касается цвета ?
изменить y на -2
если костюм № = 4 , то
1
иначе
2

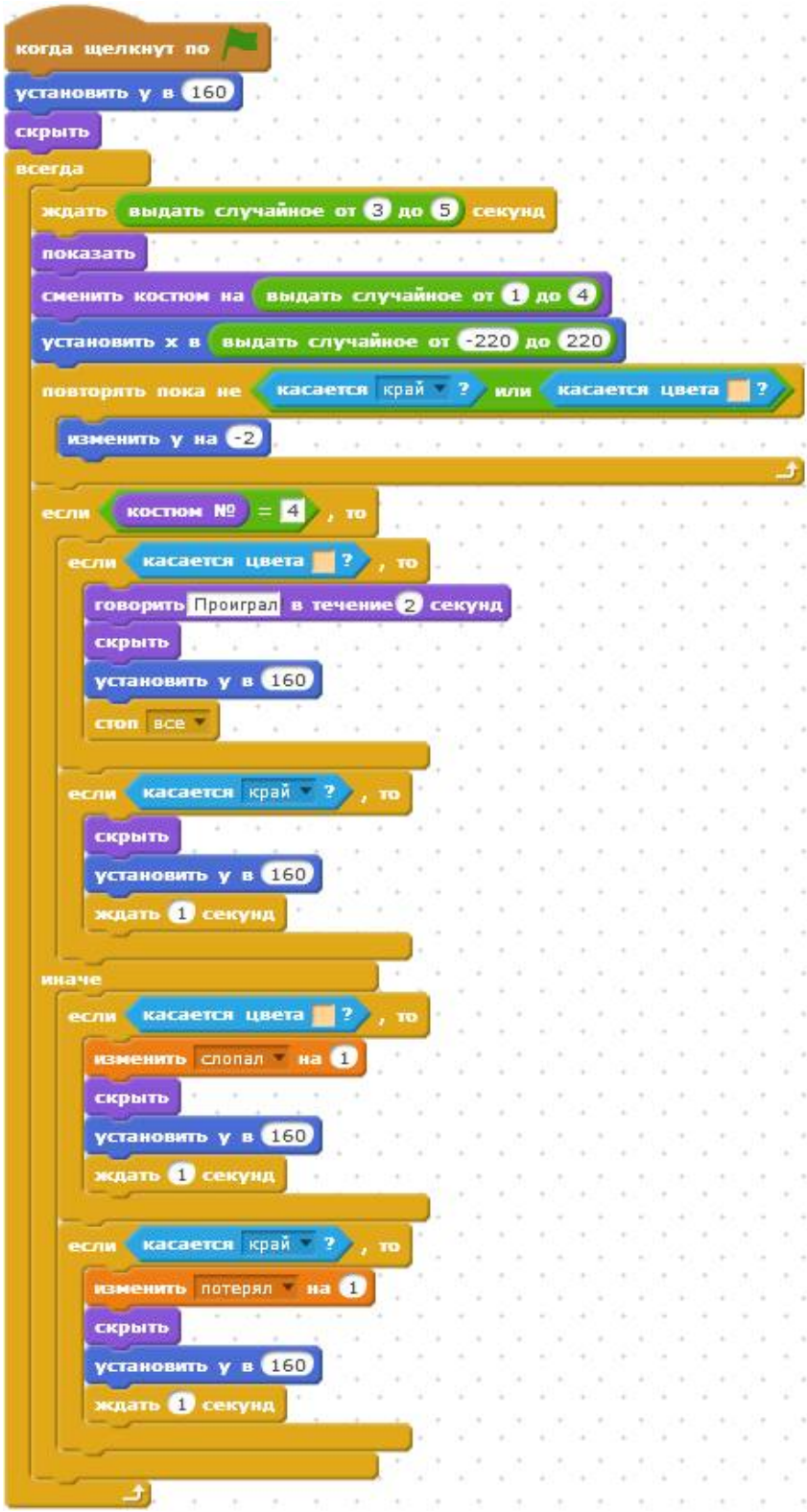
Вместо блока под номером 1 вставьте следующую часть скрипта.



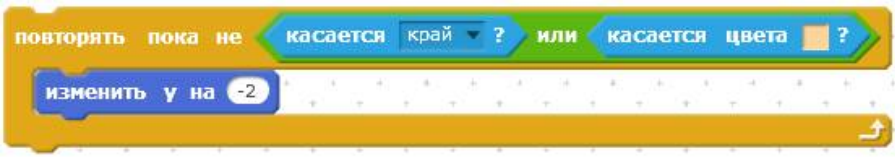
Вместо блока под номером 2 вставьте следующую часть скрипта.



Готовый скрипт будет выглядеть вот так.


когда щелкнут по
установить y в 160
скрыть
всегда
ждать выдать случайное от 3 до 5 секунд
показать
сменить костюм на выдать случайное от 1 до 4
установить x в выдать случайное от -220 до 220
повторять пока не касается край ? или касается цвета ?
изменить y на -2
если костюм № = 4 , то
если касается цвета ? , то
говорить Проиграл в течение 2 секунд
скрыть
установить y в 160
стоп все
если касается край ? , то
скрыть
установить y в 160
ждать 1 секунд
иначе
если касается цвета ? , то
изменить слопал на 1
скрыть
установить y в 160
ждать 1 секунд
если касается край ? , то
изменить потерял на 1
скрыть
установить y в 160
ждать 1 секунд

Давайте посмотрим, как он работает. Вкусняшки всегда падают сверху, поэтому вс1 время в этом скрипте мы будем возвращать их наверх блоком «установить Y в 160». Скрипт состоит из огромного блока «всегда», в котором сначала есть задержка появления следующей вкусняшки от 3 до 5 секунд. Затем происходит выбор вкусняшки, случайным образом выбирается один из четырех костюмов, а также случайным образом выбирается место падения. Затем в блоке «повторять пока не» программируется падение вкусняшек вниз. Каждая вкусняшка опускается на два шага вниз, изменяя Y на -2, до тех пор, пока не коснётся нижнего края экрана, или розового рта Кота.



Затем следует огромный блок «если иначе», в котором обрабатывается касание вкусняшками рта Кота, или края экрана. На самом деле, в этом блоке нет ничего сложного. Вот посмотрите, если Перец (номер костюма равен 4) коснётся рта Котика (розового цвета), то наш герой проиграл… Если Перец коснётся края экрана, то он сначала исчезнет, а потом снова начинает падение, установив Y в 160 (верх сцены). Вторая половина блока «если или» отвечает за движение съедобных вкусняшек. Если одна из них коснётся ротика нашего Котика, то переменная «слопал» увеличится на единицу, и это хорошо. А если вкусняшка пролетит мимо рта, и упадёт на пол, то она просто исчезнет, и появится снова на самом верху (координата Y = 160). Вот такая простая игра.



Мистер Кот, выполняйте задание: вкусняшки ловить, перец не есть, и всегда следовать за мышкой!

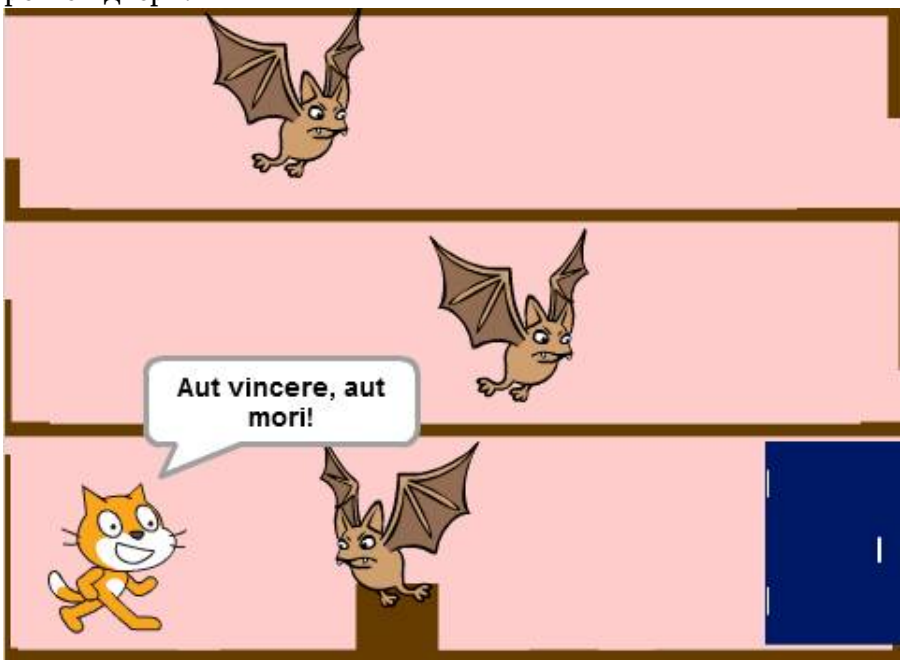
Вопрос. Что измениться в работе программы, если вместо перца нарисовать бутылочку с ядом?



Задание. Ускорьте падение вкусняшек. Сделайте так, чтобы с неба падало больше вкусняшек.

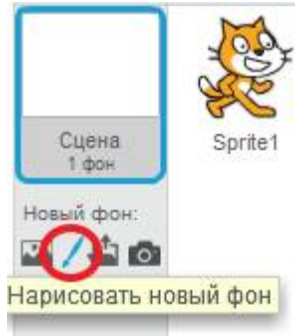
ГЛАВА 17. AUT VINCERE, AUT MORI. ПОБЕДА ИЛИ СМЕРТЬ!

Сделаем игру про приключения древнеримского Котёнка. Это был очень смелый Котёнок. Как-то раз он попал в дом с летучими мышами, насилу выбрался. «Победа или смерть!» - кричал Котёнок на латинском языке, и снова отправлялся к заветной третьей двери.

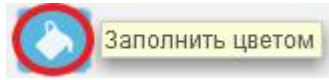


На этой главе мы познакомимся с переименованием спрайтов. В предыдущих проектах спрайтов было немного, и мы оставляли их имена без изменения. В этом проекте будет восемь спрайтов, и для того, чтобы в них не запутаться, мы дадим им нормальные понятные имена. В проекте будут использованы новые блоки «сменить фон» и «стоп». Также мы будем много рисовать в битовом режиме, познакомимся с новыми приёмами рисования и с кнопкой уменьшения размера спрайта.

Создайте новый проект. Нарисуйте три фона для сцены. Это будут три уровня нашей игры.



Сначала залейте фон розовым цветом.
Выберите инструмент «Заполнить цветом».



Выберите розовенький.



Кликните по фону, он окраситься в розовый.

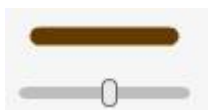
Затем выберите инструмент Линия.



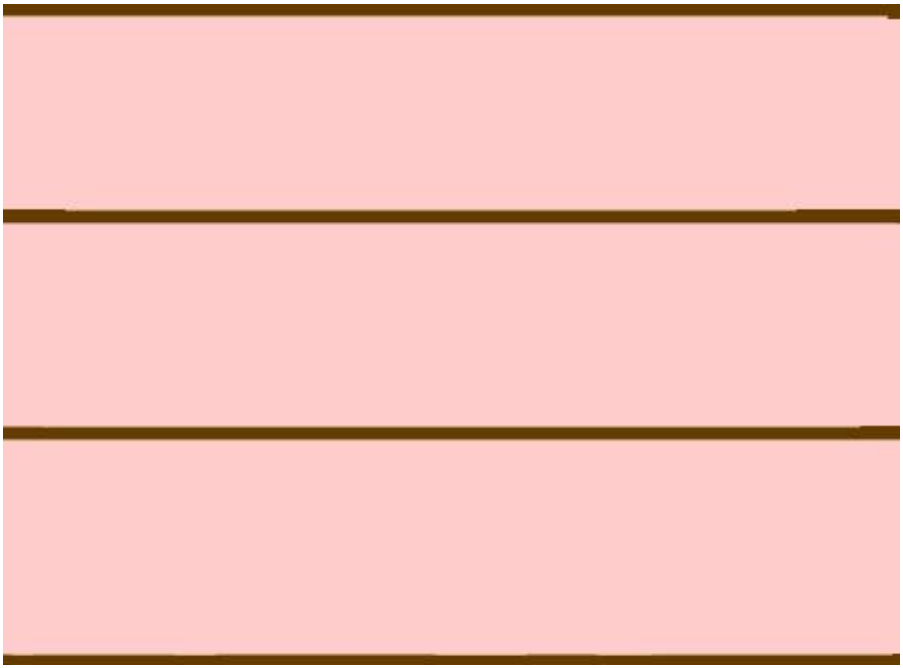
Возьмите коричневый цвет.



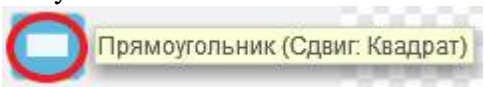
Увеличьте толщину линии.



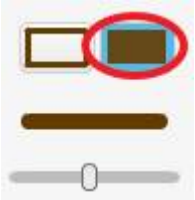
Рисуйте линии, удерживая клавишу Shift нажатой, так линии будут получаться ровнее. Нарисуйте пол, потолок и ещё две линии, делящие уровень ровно на три части.



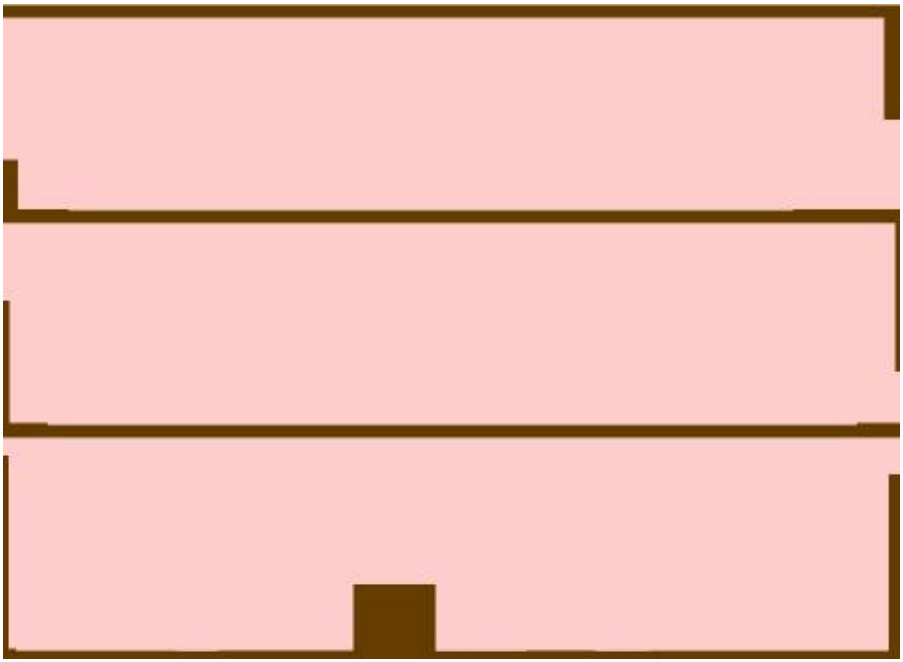
Теперь нарисуем несколько препятствий. Выберите инструмент «Прямоугольник».



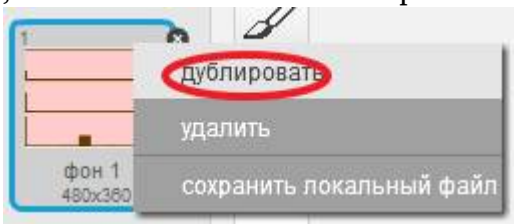
Выберите закрашенный прямоугольник.



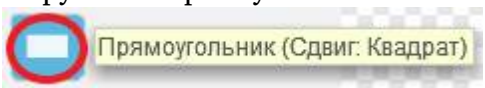
Нарисуйте препятствия.



За основу второго уровня возьмём первый уровень. Дублируем его, а затем немного изменим препятствия.



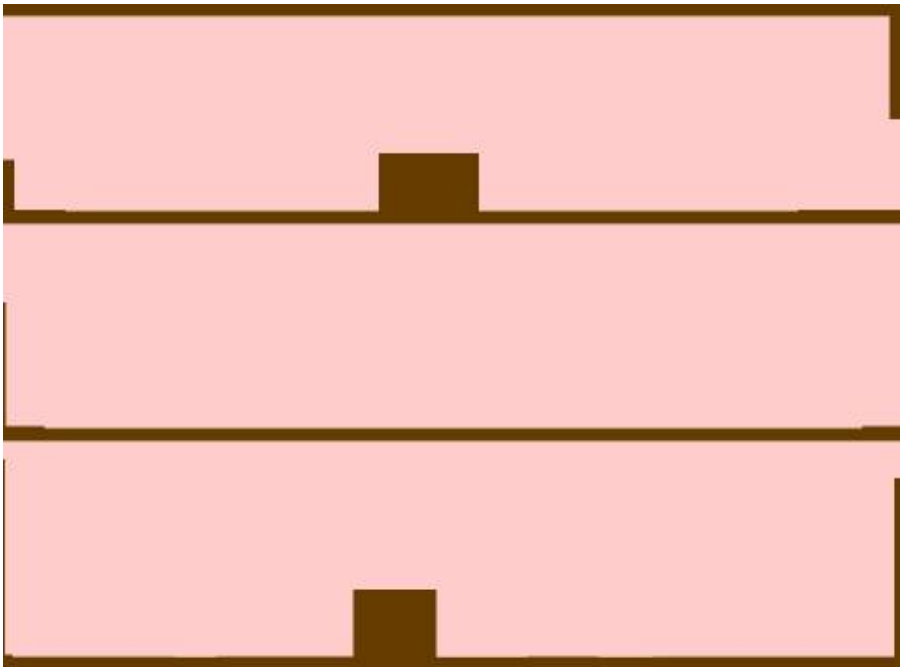
Выберите инструмент «Прямоугольник».



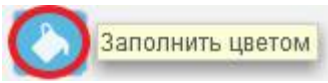
Выберите закрашенный прямоугольник.



Нарисуйте на втором уровне новые препятствия.



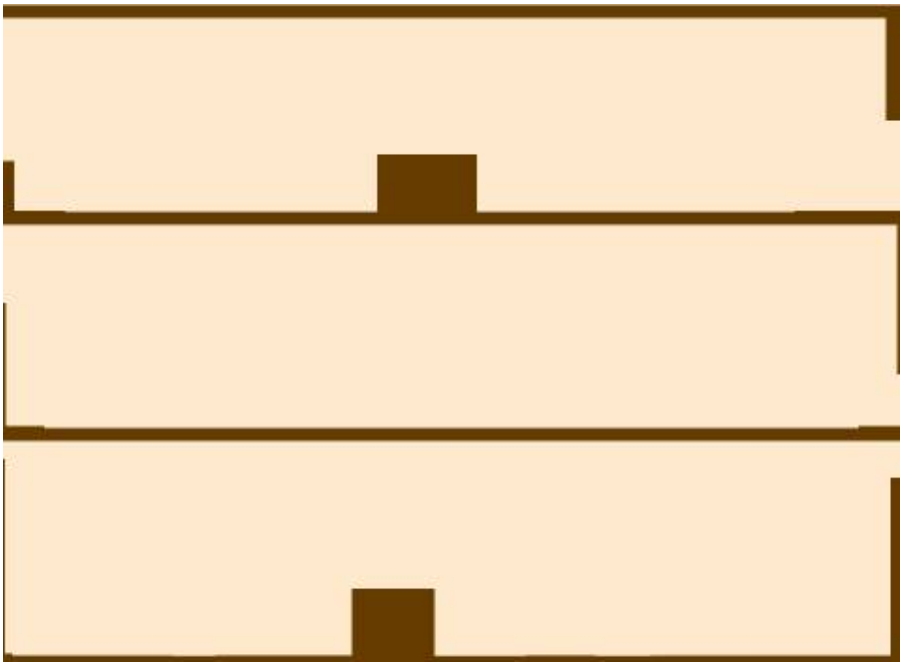
Теперь перекрасьте уровень. Выберите инструмент «Заполнить цветом».



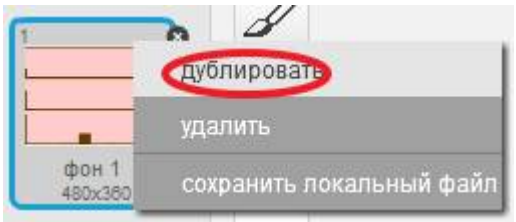
Выберите рыженький.



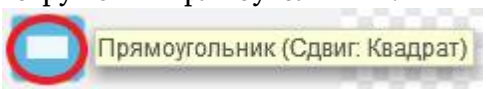
Кликайте а розовый фон, он перекраситься. Второй уровень станет таким.



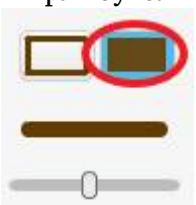
За основу третьего уровня также возьмём первый уровень. Дублируем его, а затем снова нарисуем препятствие и перекрасим.



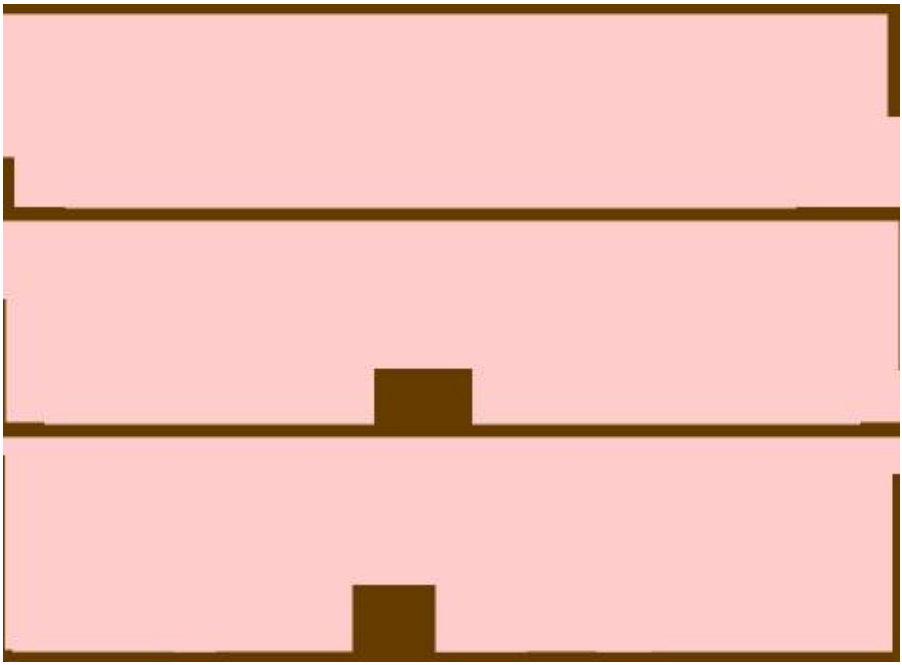
Выберите инструмент «Прямоугольник».



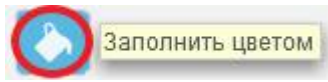
Выберите закрашенный прямоугольник.



Нарисуйте на третьем уровне новое препятствие.



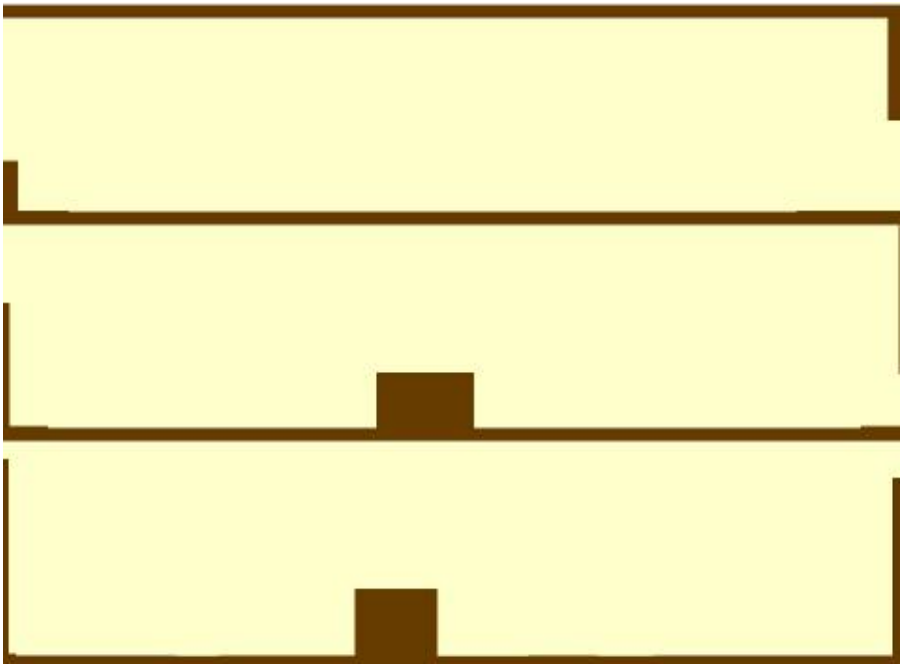
Теперь перекрасьте уровень. Выберите инструмент «Заполнить цветом».



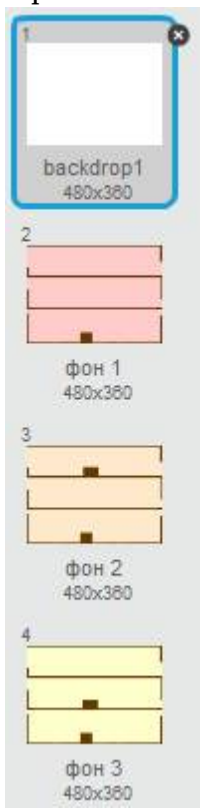
Выберите жёлтенький.



Кликайте а розовый фон, он перекраситься. Третий уровень станет таким.



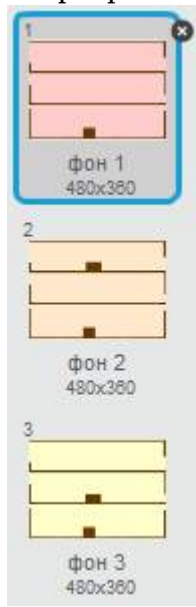
Теперь у Сцены четыре фона.



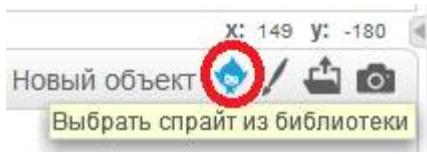
Первый фон пустой, его надо удалить. Нажмите на маленький крестик.



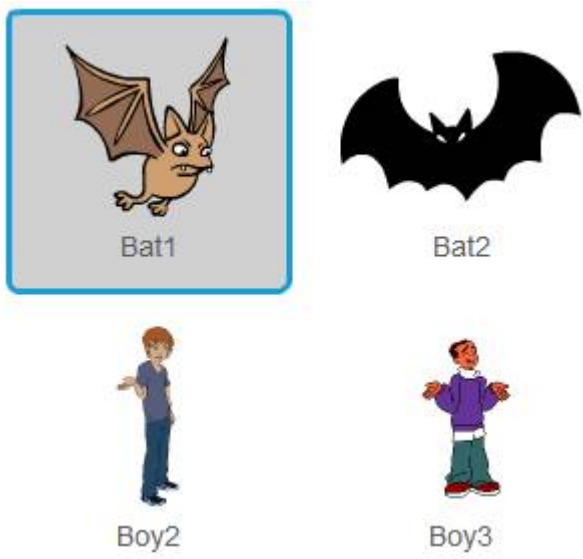
Должно остаться только три фона.



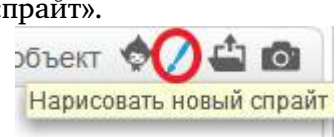
Теперь надо добавить врагов, которые будут бегать по уровням. Выберите из библиотеки одного врага. После того, как мы запрограммируем его, то с помощью дублирования сделаем ещё двоих врагов.



Пусть врагом будет летучая мышь.



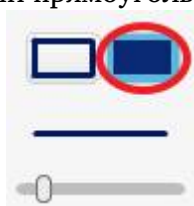
Для перехода с уровня на уровень нам понадобится дверь. Нарисуйте её в графическом редакторе. Нажмите на кнопку «Нарисовать новый спрайт».



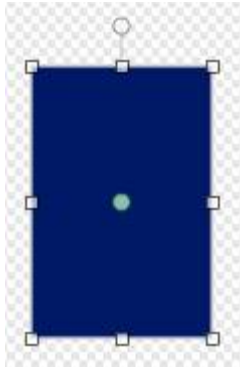
Выберите инструмент «Прямоугольник» и тёмно-синий цвет.



Выберите закрашенный прямоугольник.



Нарисуйте прямоугольник.



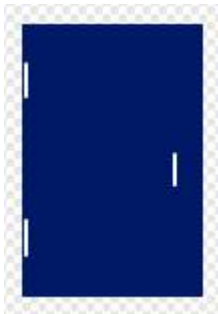
Теперь выберите белый цвет и подрисуйте дверную ручку и петли.



Выберите инструмент Линия.



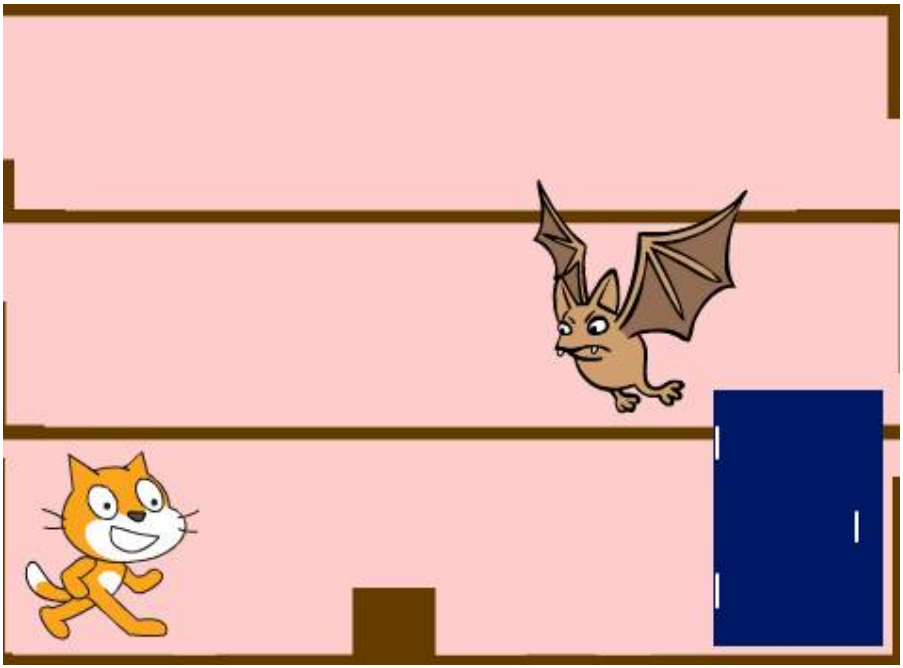
Проведите несколько коротких линий, удерживая клавишу Shift.



Вторую и третью дверь сделаем после того, как запрограммируем первую дверь с помощью дублирования.



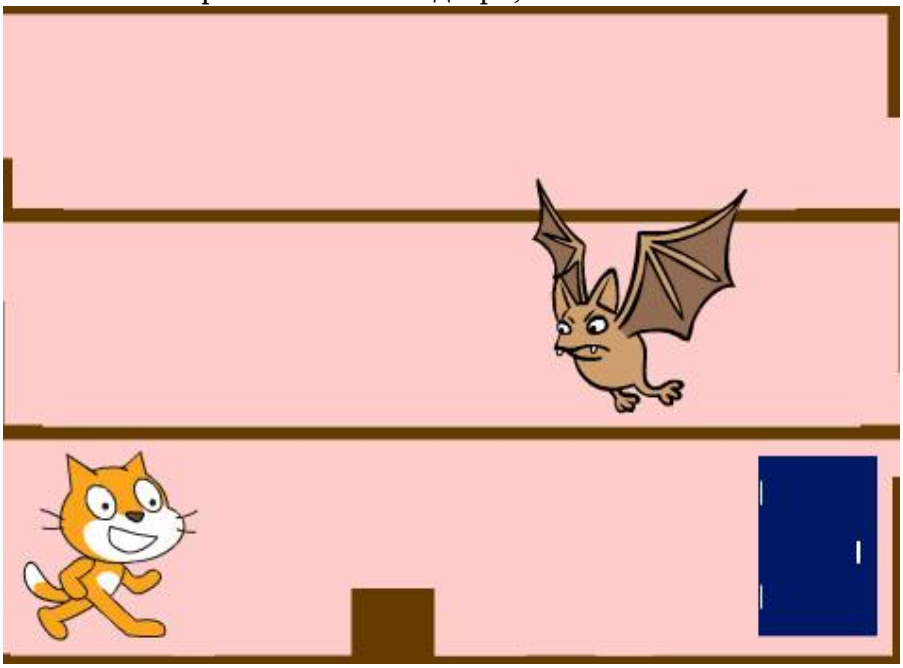
Совет. Если прямоугольник получился очень большой, и, как у меня, не помещается на этаже, то уменьшить его можно с помощью специальной кнопочки.



Кликните на кнопку, а потом на дверь.

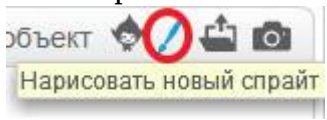


Несколько раз кликните по двери, пока она не поместится.

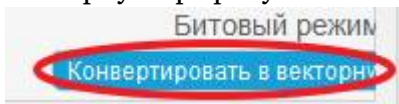


Теперь нарисуйте скрипт Победа. Он будет появляться в

конце игры, когда Кот пройдёт все три уровня. Нажмите на кнопку «Нарисовать новый спрайт».



Перейдите в векторный режим, нажав на кнопку «Конвертировать в векторную графику».



Нажмите на кнопку «Текст».



Выберите цвет текста, например, такой.



Напишите «Gloria victoribus!» что в переводе с латыни означает «Слава победителям!»



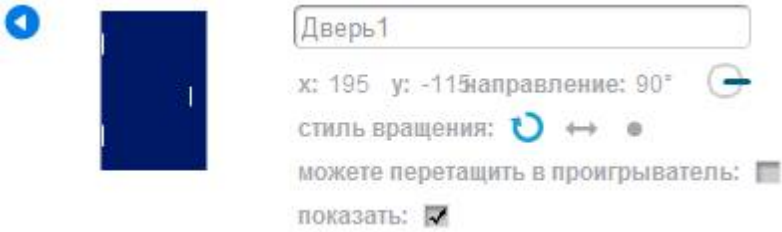
Начинаем программировать поведение спрайтов, а также переименовывать их. Начнём с первой двери. Выберите её и нажмите на синий кружочек.



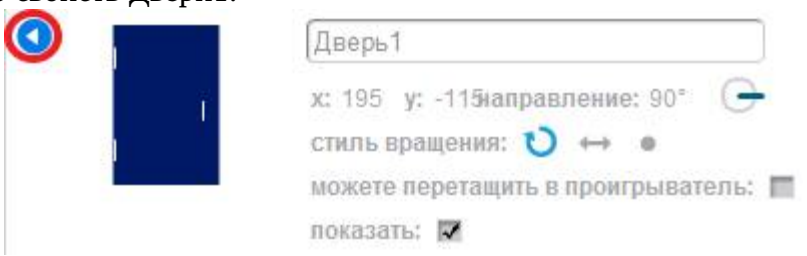
Откроется окно свойств спрайта.



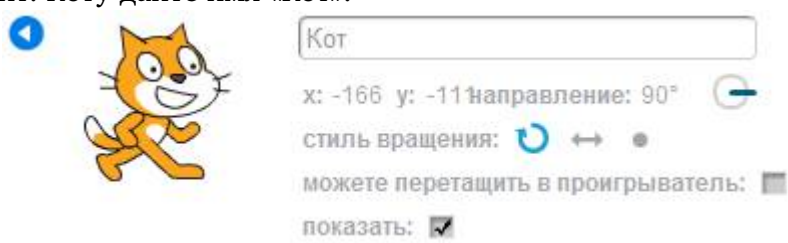
Сейчас имя спрайта Sprite2, измените его на Дверь1.



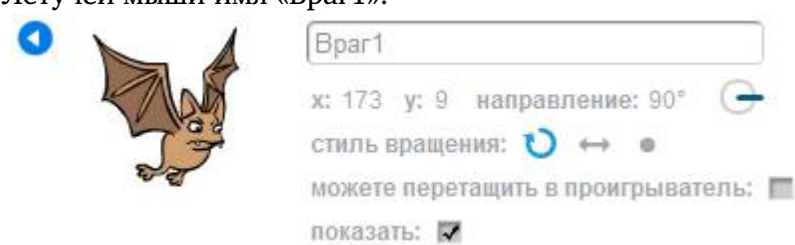
Нажмите на синий кружочек со стрелочкой чтобы закрыть окно свойств Двери1.



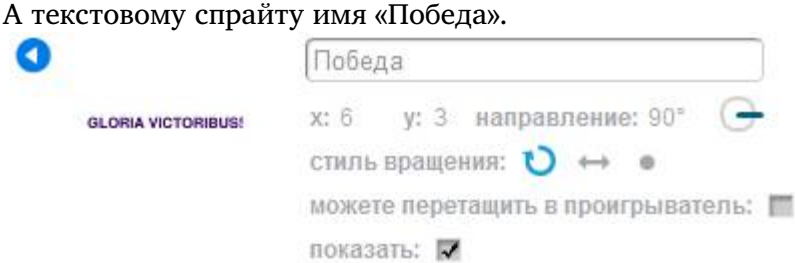
Сразу переименуйте Кота, летучую мышь, и текстовый спрайт. Коту дайте имя «Кот».



Летучей мыши имя «Враг1».

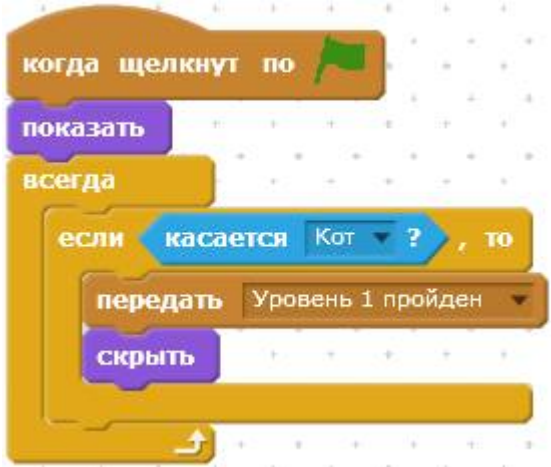


А текстовому спрайту имя «Победа».



Начинаем программировать спрайты. Сначала сделаем программу для первой двери. Она будет состоять из двух скриптов. Сделайте первый скрипт Двери1. Он работает очень просто. Когда Кот дойдёт до двери и коснётся её, то будет

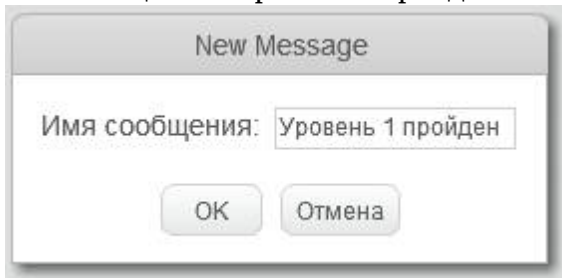
передано сообщение о прохождении первого уровня и дверь скроется.



Помните, как создавать сообщения? Кликните на чёрный треугольничек и выберите «новое сообщение».



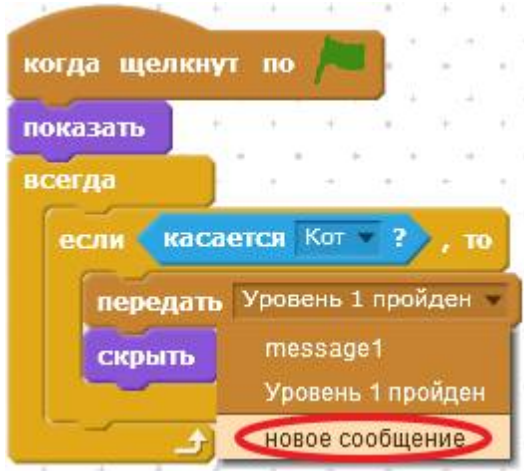
Введите имя сообщения «Уровень 1 пройден».



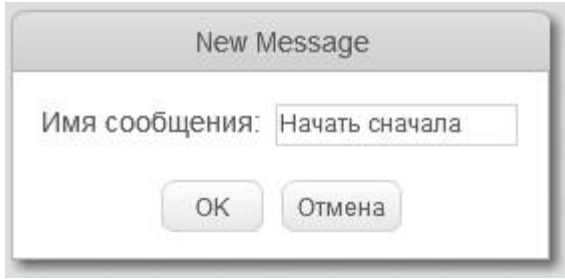
Затем создайте второй скрипт для Двери1.



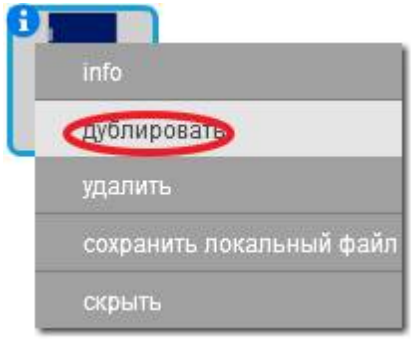
Для того, чтобы создать сообщение «Начать сначала, кликните на маленький чёрный треугольничек в блоке «передать».



Напишите название события «Начать сначала».



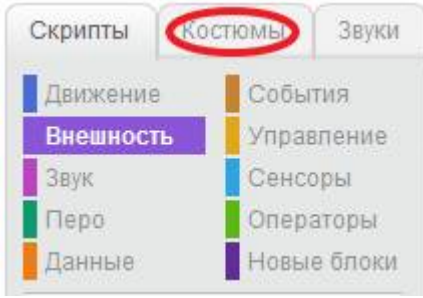
Сообщение «Начать сначала» передаётся в начале работы программы, а также всякий раз, когда Кот проиграет и коснётся Врага или коричневой преграды. Первая дверь готова, дублируйте её.



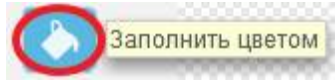
Новая дверь автоматически получила имя Дверь2.



Давайте перекрасим её в другой цвет. Выберите Дверь2 и кликните на вкладку Костюмы.



Выберите инструмент «Заполнить цветом».



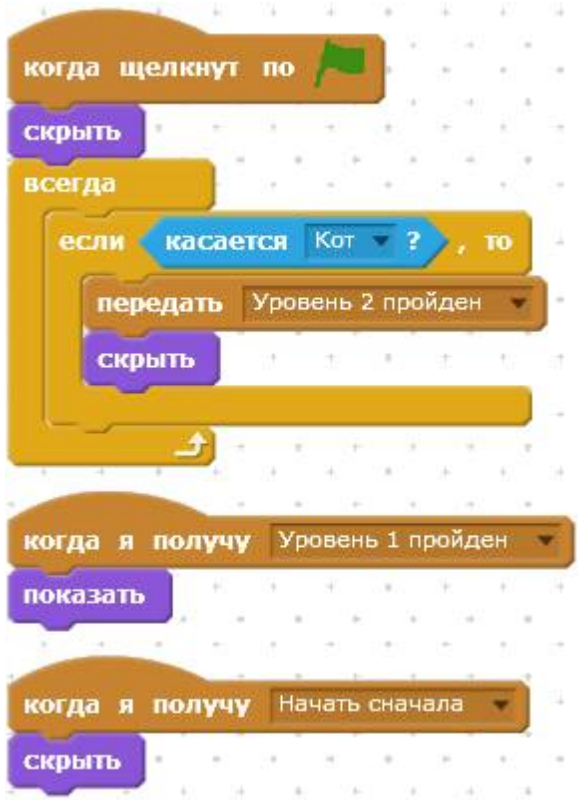
Выберите какой-нибудь цвет, например, такой.



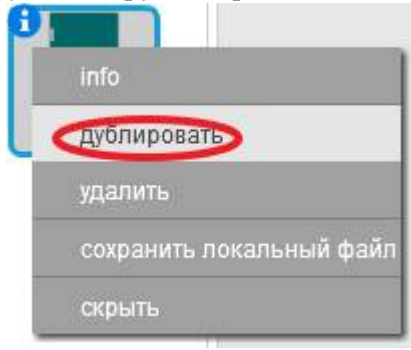
Кликните в дверь, она перекрасится.



Теперь перейдите на вкладку «Скрипты» и сделайте такую программу Двери2.



Теперь дублируйте вторую дверь.



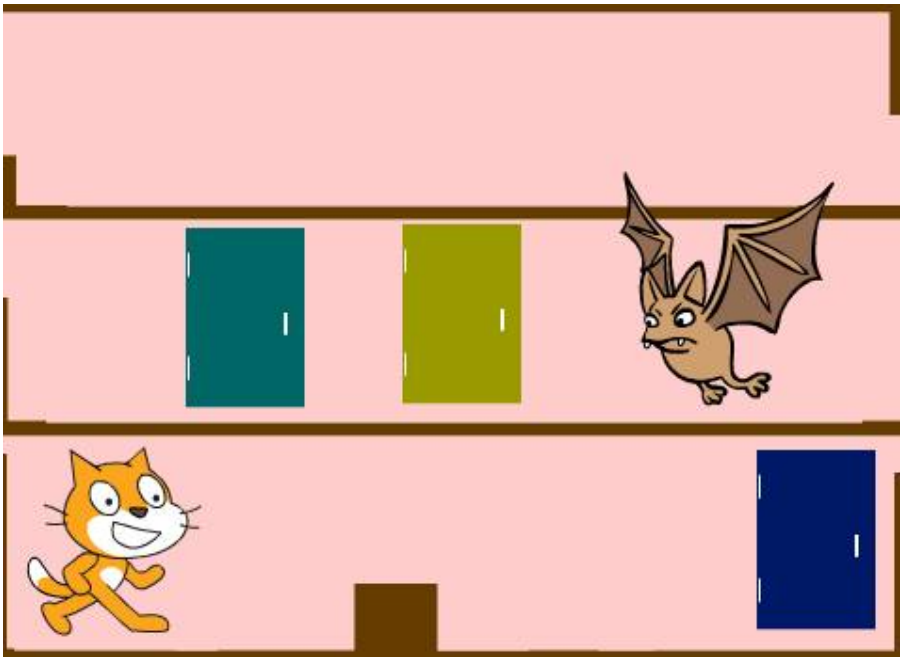
Перекрасьте её.



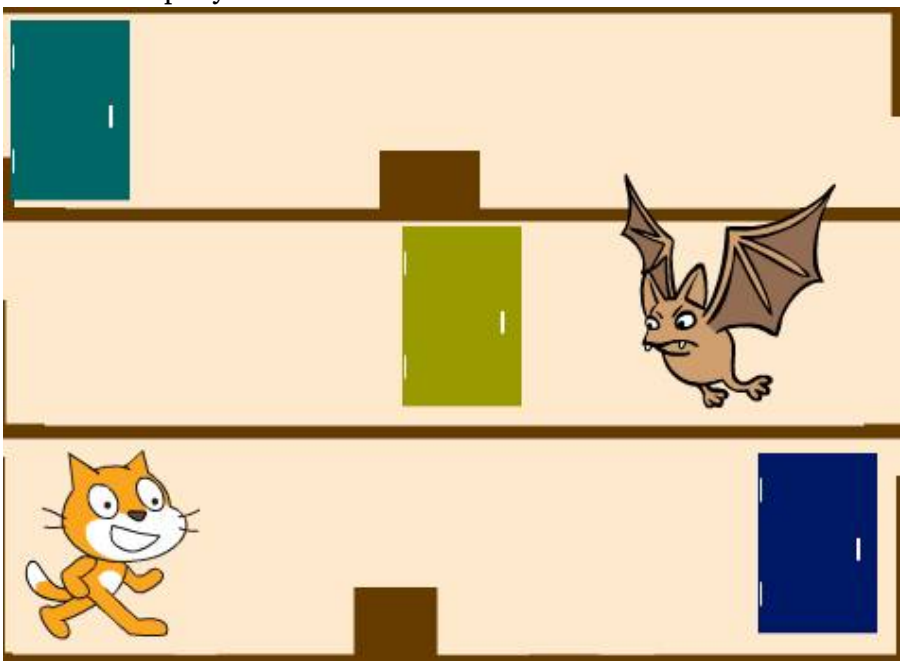
Сделайте для третьей двери такую программу.



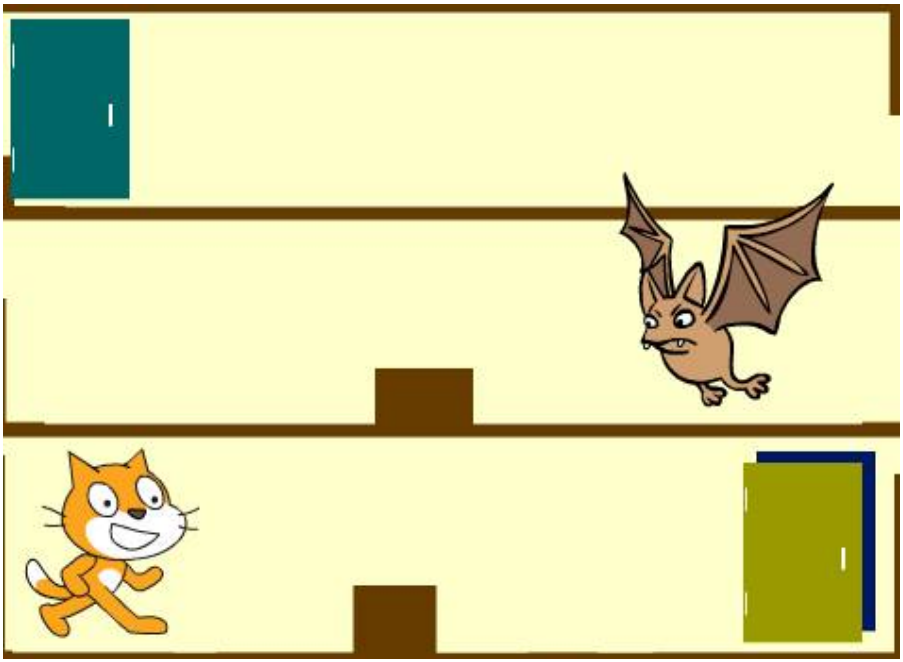
Программы дверей готовы, теперь их надо разместить на своих местах. Выберите Сцену и нажмите вкладку «Фоны». Выберите первый фон и разместите первую дверь как показано на рисунке, справа внизу.



Теперь выберите фон2 и разместите Дверь2 слева вверху, как показано на рисунке.



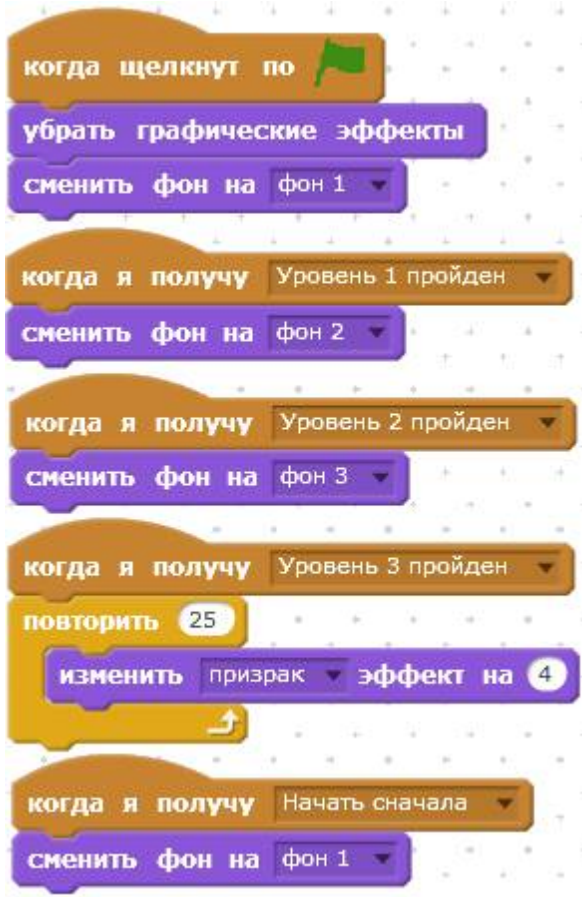
Переключите Сцену на третий фон и поместите третью дверь справа внизу, там же где и первая дверь.



Если какая-либо дверь имеет очень большой размер, то уменьшите её специальной кнопкой.



Теперь сделайте программу для Сцены.


когда щелкнут по
убрать графические эффекты
сменить фон на фон 1
когда я получу Уровень 1 пройден
сменить фон на фон 2
когда я получу Уровень 2 пройден
сменить фон на фон 3
когда я получу Уровень 3 пройден
повторить 25
изменить призрак эффект на 4
когда я получу Начать сначала
сменить фон на фон 1

Здесь использованы новые блоки «сменить фон на фон». В начале игры виден фон1. После того как Кот пройдёт первый уровень, будет передано сообщение «Уровень 1 пройден» и фон сменится на фон2. После прохождения второго уровня фон смениться на фон2, а после прохождения третьего уровня фон плавно раствориться из-за применения эффекта «призрак». Эффект 25 раз измениться на 4 процента, всего на 100 процентов, то есть фон станет полностью прозрачным.

А сейчас сделайте программу для Врага1. Первые три скрипта.



Четвёртый скрипт Врага1.

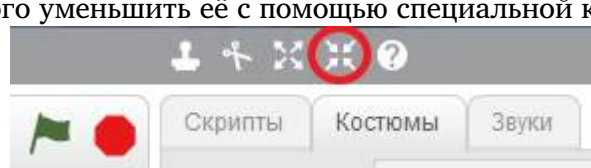


Программа Врага не очень сложная. Когда он получает сообщение «Начать сначала», от он идёт в начальную точку на левом краю сцены, а затем двигается направо в направлении 90

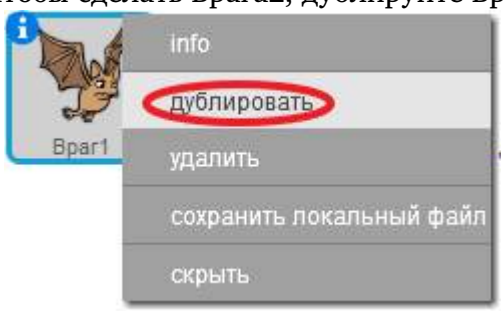
градусов изменяя координату X на единицу 480 раз подряд. Таким образом летучая мышь долетает до правого края экрана, до X равного 240. Затем она разворачивается, меняя направление на -90 градусов и двигается влево, изменяя X на -1. Вот так она «всегда» и летает, до тех пор, пока не получит сообщение «Уровень 3 пройден». В этот момент летучая мышь скроется и сработает новый блок «стоп другие скрипты спрайта». В результате будет остановлена работа всех скриптов Врага1, в том числе и блоков «всегда».



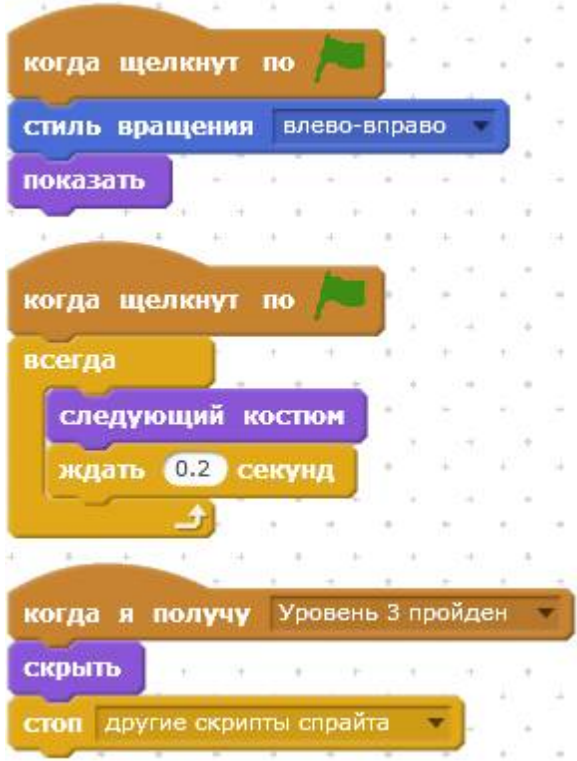
Совет. Наша летучая мышь не помещается на этаже целиком, надо немного уменьшить её с помощью специальной кнопки.



Для того, чтобы сделать Врага2, дублируйте Врага1.



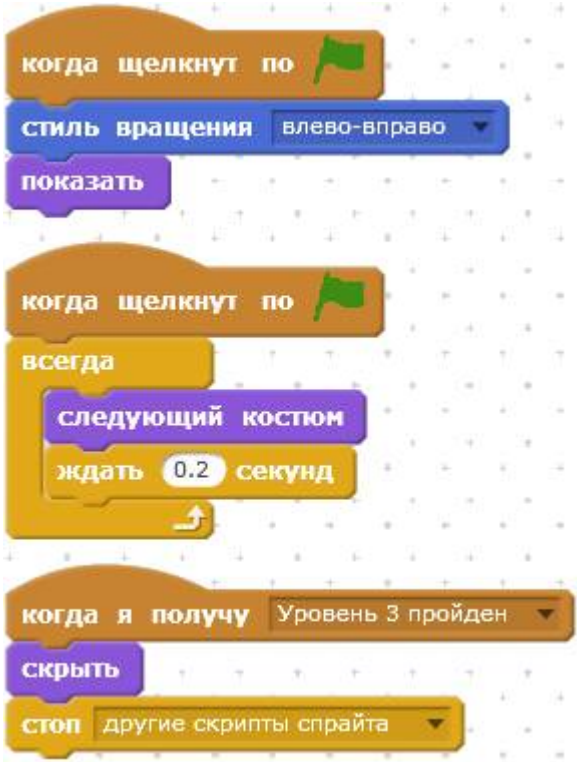
Сделайте ему такую программу. Первые три скрипта.



Четвёртый скрипт Врага2.



Снова дублируйте Врага1, получится Враг3. Измените его программу. Вот первые три скрипта.



Вот четвёртый скрипт Врага3.



Протестируйте работу скриптов. Нажмите на зелёный флажок и посмотрите, как летают летучие мыши. Нижняя летучая мышь

должна лететь справа налево, средняя слева направо, а верхняя слева направо с задержкой в три секунды. Если у Вас враги летают не так, то внимательно посмотрите на скрипты и исправьте ошибки. Если все летают правильно, то можно сделать программу для Кота. Но для начала немного уменьшите Котика, так ему будет проще пройти все уровни. Какой кнопкой его уменьшить Вы должны помнить. Всего программа Кота состоит из восьми скриптов. Вот первые два скрипта.

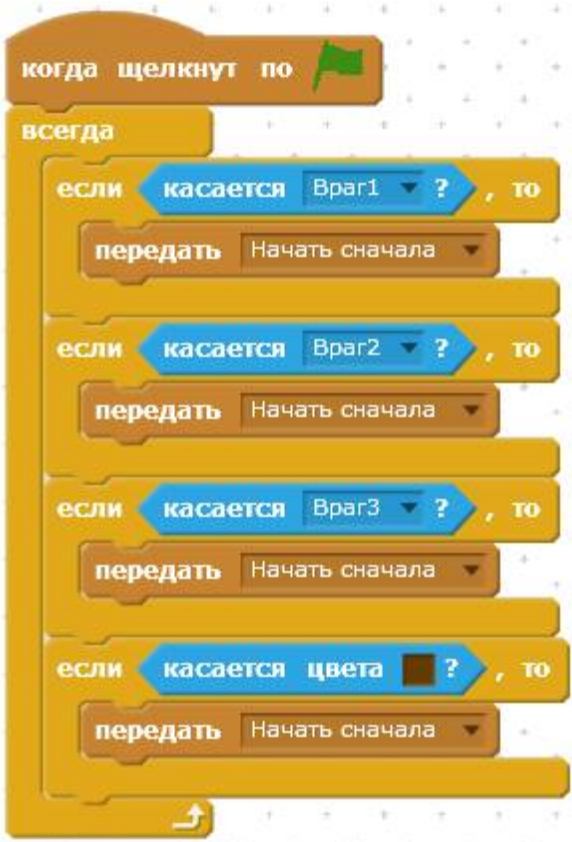

когда щелкнут по
стиль вращения влево-вправо
показать
передать Начать сначала
когда я получу Начать сначала
перейти в x: -180 y: -120
играть звук meow
говорить Aut vincere, aut mori! в течение 5 секунд

Это начальные скрипты. Как видите, именно Кот передаёт первое сообщение «Начать сначала» и тут же сам получает его, переходит в левый нижний угол, мяукает и говорит серьёзную фразу «Победа или смерть!».



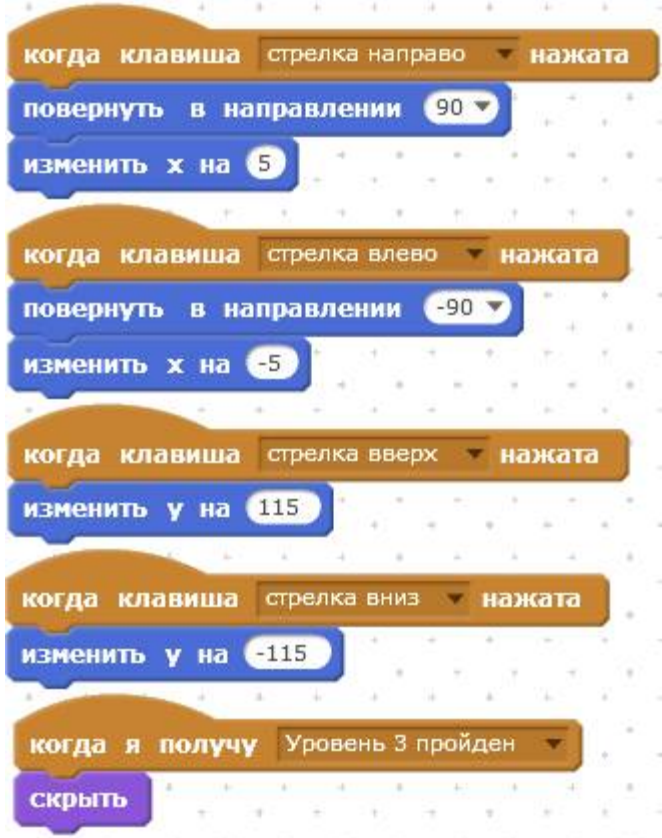
В упорстве нашему Котику не откажешь!

Третий скрипт Кота такой.


когда щелкнут по
всегда
если касается Враг1 ? , то
передать Начать сначала
если касается Враг2 ? , то
передать Начать сначала
если касается Враг3 ? , то
передать Начать сначала
если касается цвета ? , то
передать Начать сначала

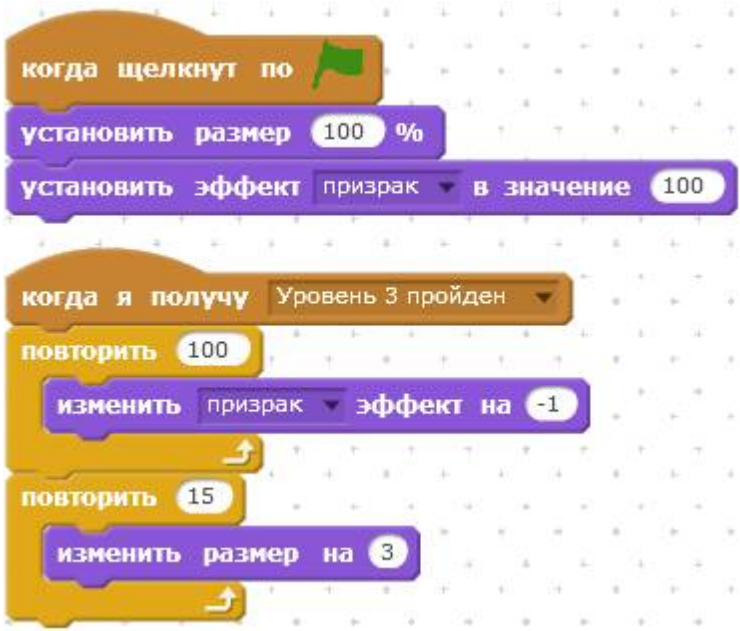
Здесь всё тоже очень просто. Если Кот коснётся любого Врага или коричневого цвета, то игра начнётся сначала.

Следующие пять скриптов Кота — это скрипты управления его движением с помощью стрелок.


когда клавиша стрелка направо нажата
повернуть в направлении 90
изменить x на 5
когда клавиша стрелка влево нажата
повернуть в направлении -90
изменить x на -5
когда клавиша стрелка вверх нажата
изменить y на 115
когда клавиша стрелка вниз нажата
изменить y на -115
когда я получу Уровень 3 пройден
скрыть

Последний скрипт Кота скрывает его в конце игры, когда пройден третий уровень.

Теперь сделайте программу победному скрипту.


когда щелкнут по
установить размер 100 %
установить эффект призрак в значение 100
когда я получу Уровень 3 пройден
повторить 100
изменить призрак эффект на -1
повторить 15
изменить размер на 3

Запустите программу. Попробуйте пройти все уровни.



Вопрос. 1. Как изменить скорость Кота?
2. Как изменить скорость Летучих мышей?
Задание. Добавьте в игру ещё один уровень.

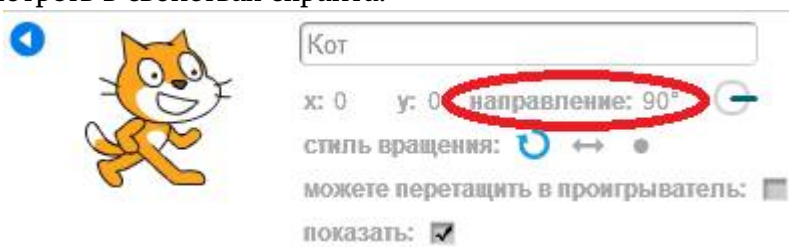
ПРИЛОЖЕНИЕ 1. ОСНОВНЫЕ ПОНЯТИЯ SCRATCH 2.

В этом приложении кратко объяснены основные понятия Scratch 2:

- Направление;
- Выпадающий список;
- Стили вращения;
- Пронумерованные цвета;
- Графические эффекты;
- Слои.

А. НАПРАВЛЕНИЕ.

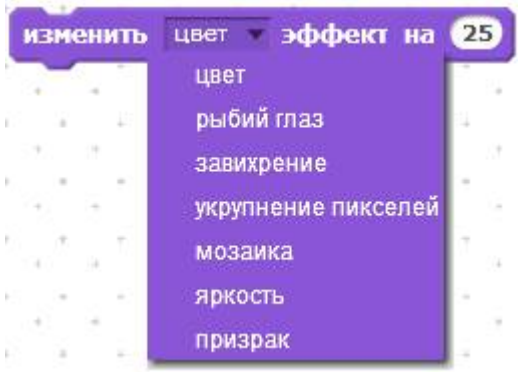
Под направлением спрайта понимается направление движения спрайта, в котором он будет перемещаться при использовании блока «Идти». Текущее направление можно посмотреть в свойствах спрайта.



Направление измеряется в градусах. Направлению в ноль градусов соответствует движение вертикально вверх. Направление 90 градусов — движение вправо, направление -90 градусов — движение влево. Направление 180 градусов — движение вниз. Подробнее о градусах можно узнать в приложении 3.

Б. ВЫПАДАЮЩИЙ СПИСОК.

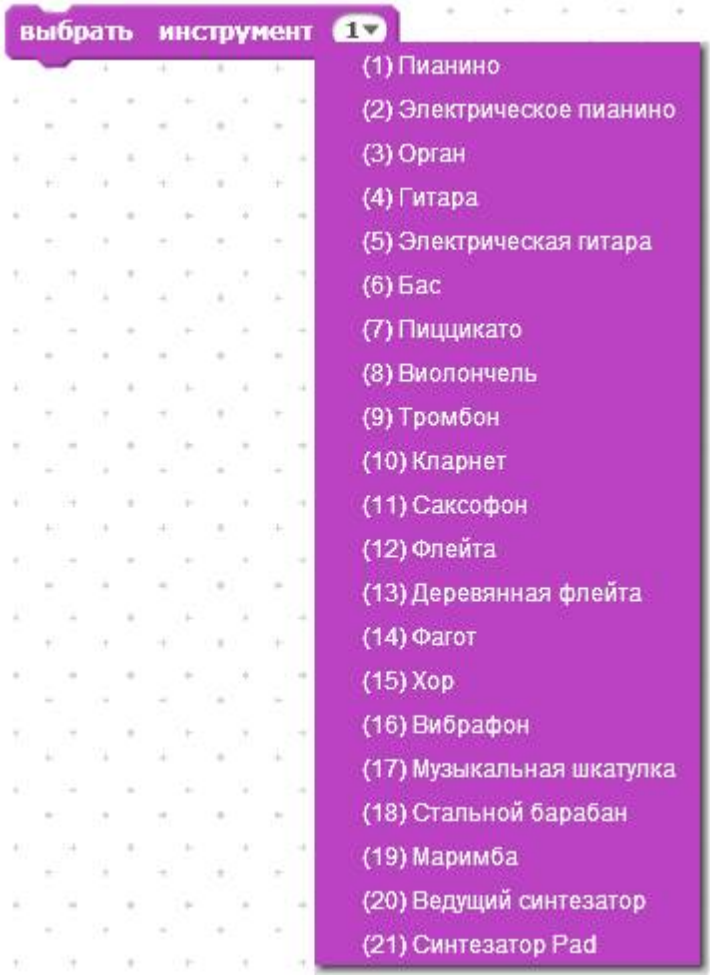
Некоторые блоки позволяют выбирать варианты с помощью выпадающего списка. Например, блок изменения эффектов позволяет выбрать один из семи эффектов.



Блок управления спрайтом с клавиатуры позволяет выбрать, на какую клавишу реагировать. Выпадающий список очень длинный, включает в себя все буквы английского алфавита, цифры, стрелки и пробел.



Блок выбора музыкального инструмента позволяет выбрать один из двадцати одного музыкального инструмента

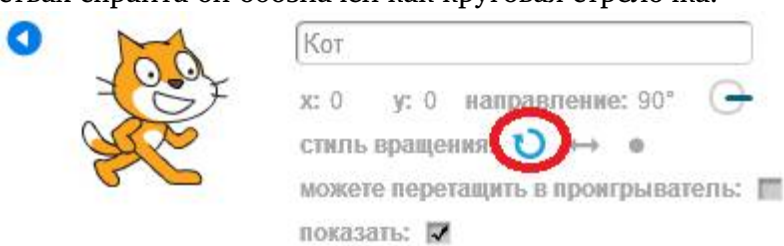


В. СТИЛИ ВРАЩЕНИЯ.

В Scratch 2.0 существует всего три типа вращения. Рассмотрим каждый из них.

Круговой стиль вращения.

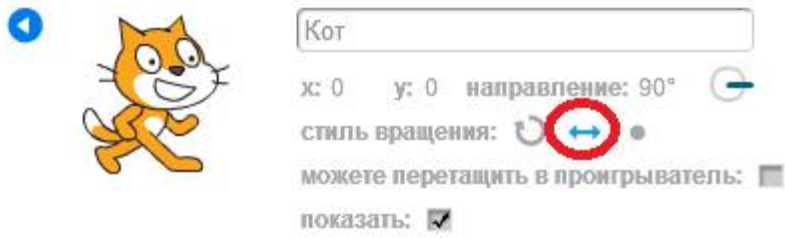
Спрайт может поворачиваться в любом направлении. В свойствах спрайта он обозначен как круговая стрелочка.



При изменении текущего направления спрайта, спрайт будет поворачиваться на требуемое количество градусов.

Стиль вращения вправо-влево.

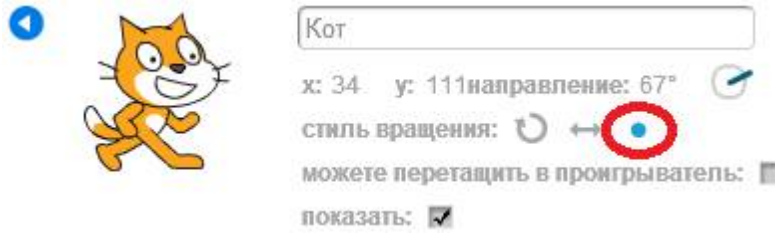
Спрайт может поворачиваться только в двух направлениях. Направо или налево. Поворот в остальных направлениях невозможен. В свойствах спрайта он обозначается как двойная стрелочка.



При изменении текущего направления спрайта, спрайт не будет поворачиваться на требуемое количество градусов, как в круговом стиле вращения. При текущем направлении спрайта от 0 до 180 градусов спрайт будет повернут вправо, при направлении от 0 до -180 градусов, спрайт будет повёрнут влево.

Не вращать.

При выборе этого стиля вращения спрайт не будет вращаться. В свойствах спрайта он обозначен как кружочек.



При изменении текущего направления спрайта вращаться он не будет.

Г. ПРОНУМЕРОВАННЫЕ ЦВЕТА И ОТТЕНКИ.

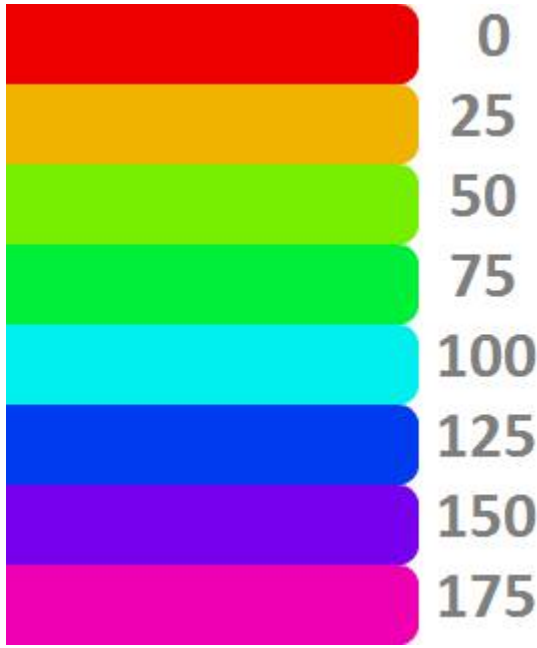
В Scratch каждый цвет имеет свой номер. Оттенки цвета также имеют номера. Это сделано для более удобной работы с цветом. Числовые значения цветов используются в блоках «изменить цвет пера» и «установить цвет пера».



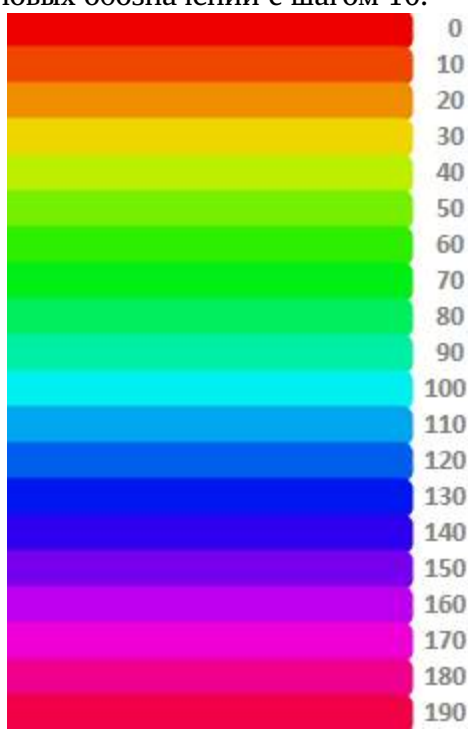
Числовые значения оттенков используются в блоках ««изменить тень пера» и «установить тень пера». Обратите внимание, из-за ошибки перевода на русский язык оттенок пера назван тенью, возможно, это скоро будет исправлено.



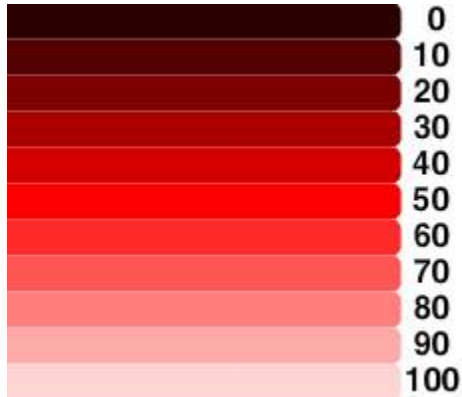
Началом отсчёта, цветом с номером ноль, как и в радуге, является красный цвет. На следующем рисунке приведена таблица соответствия цветов и их числовых обозначений с шагом 25.



Ниже приведена более подробная таблица соответствия цветов и их числовых обозначений с шагом 10.



Все оттенки цветов также пронумерованы от ноля, самого тёмного оттенка, до ста, самого светлого оттенка. Ниже приведена таблица оттенков. Нормальный красный цвет получается при значении оттенка в 50.



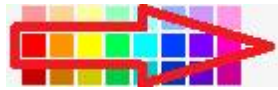
Д. ГРАФИЧЕСКИЕ ЭФФЕКТЫ.

В Scratch 2.0 есть семь графических эффектов, которые могут быть применены к спрайтам и фонам. Графические эффекты применяются с использованием блоков «Изменить эффект на», и «Установить эффект в значение». Вид графического эффекта выбирается с помощью выпадающего списка, значение эффекта вводится с клавиатуры. Блок «Убрать графические эффекты», как следует из его названия, убирает все графические эффекты.

Эффект «цвет».



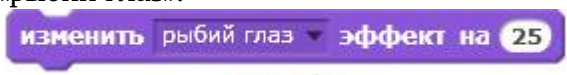
Это циклический эффект. Цвет спрайта изменяется при изменении значения эффекта от 0 до 200. При значении больше 200, например, при 201, спрайт выглядит так же, как и при 1. При 300, так же, как и при 100. При 400, так же как при 200 и при 0, то есть, без изменения цвета. Например, если у нас есть спрайт красного цвета, то при значении эффекта «цвет» равного 0, спрайт будет красным. При значении эффекта 25, спрайт станет оранжевым. При значении эффекта 50, спрайт станет жёлтым и так далее, пока, при достижении значения эффекта 200, спрайт снова не станет красным. Направление изменения цвета спрайта показано на рисунке.



Если спрайт изначально был не красного, а, например, как Кот, оранжевого цвета, то при значении эффекта 25 он станет жёлтым, при 50 зелёным, а при 75 — голубым. Изменение цветов будет происходить в том же порядке, и, при достижении значения эффекта 200, спрайт снова станет оранжевым.

Числовые обозначения цветов также используются при рисовании пером. Ниже приведена таблица соответствия цветов и их числовых обозначений с шагом 25.

Эффект «рыбий глаз».

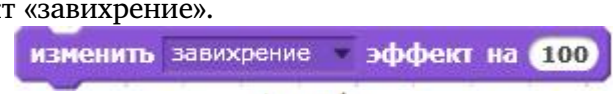




При применении этого эффекта спрайт как будто надувается изнутри. Края спрайта остаются почти без изменений, а центр увеличивается в размере. Эффект не циклический, значение эффекта может быть очень велико, однако, при значении эффекта больше 1000, спрайт практически не изменяется, он превращается в круглое цветное пятно. Самые интересные изменения происходят со спрайтом при изменении значения эффекта «рыбий глаз» от 0 до 200. На рисунке показан Кот со значением эффекта «рыбий глаз» равным 200.



Эффект «завихрение».



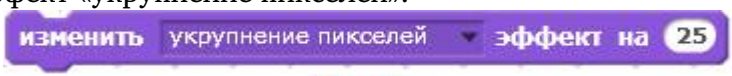


При применении этого эффекта спрайт как будто

закручивается изнутри. Края спрайта остаются почти без изменений, а центр вращается против часовой стрелки. Эффект не циклический, значение эффекта может быть очень велико, однако, при значении эффекта больше 3000, спрайт практически не изменяется, он превращается в круглое разноцветное пятно. Самые интересные изменения происходят со спрайтом при изменении значения эффекта «завихрение» от 0 до 500. На рисунке показан Кот со значением эффекта «рыбий глаз» равным 500.



Эффект «укрупнение пикселей».





Пиксели — это точки на экране из которых состоит изображение спрайта. При применении эффекта «укрупнение пикселей», точки изображения начинают увеличиваться в размерах и сливаться, образуя всё более крупные квадратики. В результате, при значении эффекта равном 500, остаётся один или два цветных квадратика.



Эффект «мозаика».





Этот эффект превращает спрайт в мозаику. Изменения происходят не плавно, а скачкообразно. Например, при значении эффекта от 0 до 4, со спрайтом ничего не происходит. При значении 5 спрайт превращается в мозаику из четырёх элементов. При дальнейшем изменении значения от 5 до 14 спрайт по-прежнему состоит из четырёх элементов, а при достижении 15 количество котиков скачкообразно увеличивается до девяти.



При достижении значения эффекта «мозаика» 25 котиков станет 16.



При достижении значения эффекта «мозаика» 35 котиков станет 25.



При дальнейшем увеличении значения эффекта котиков уже невозможно разобрать, спрайт превращается в набор цветных точек. Вот так будет выглядеть Кот при применении к нему эффекта «мозаика» со значением 200.



Эффект «яркость».





При применении этого эффекта спрайт становится ярче. При значении эффекта больше 100 никакие изменения не происходят, потому что спрайт достиг максимальной яркости. На рисунке показан Кот с яркостью 100. Все линии чёрного цвета стали белыми.



Эффект «призрак».





Этот эффект можно было бы назвать «прозрачность», он задаёт величину прозрачности спрайта. При значении эффекта «призрак» равного 0, прозрачности нет совсем, а при значении 100, прозрачность максимальна, то есть спрайт вообще невидим.

Е. СЛОИ.

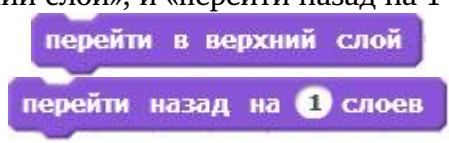
Слои определяют взаимное расположение спрайтов на сцене. Если один из спрайтов расположен в верхнем слое, а другой на один слой ниже, то спрайт, расположенный в верхнем слое будет виден целиком, а спрайт, расположенный на слой ниже, будет из-за него выглядывать. Сцена всегда находится в самом нижнем слое. В данном примере товарищ Гобо находится в верхнем слое, а гражданин Кот на слой ниже.



Так получилось из-за того, что Гобо был добавлен позднее. Каждый новый спрайт, добавленный в проект, располагается в самом верхнем слое. Добавленная последней синяя Собака расположена в третьем слое, выше, чем товарищ Гобо.



Для управления расположением спрайтов служат блоки «перейти в верхний слой», и «перейти назад на 1 слоев».



ПРИЛОЖЕНИЕ 2. СПРАВОЧНИК ПО БЛОКАМ SCRATCH 2.0.

В этом приложении перечислены все блоки Scratch 2 с кратким описанием.

А. СИНИЕ БЛОКИ ДВИЖЕНИЯ.



Спрайт переместиться на 10 шагов в текущем направлении.



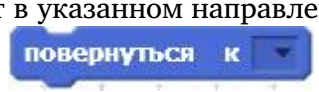
Повернуть спрайт на 15 градусов по часовой стрелке по отношению к текущему направлению спрайта.



Повернуть спрайт на 15 градусов против часовой стрелки по отношению к текущему направлению спрайта.



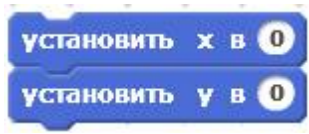
Повернуть спрайт в указанном направлении.



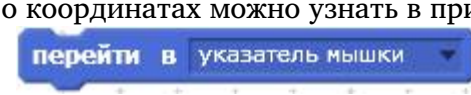
Повернуться по направлению к одному из спрайтов. Для выбора спрайта следует нажать на чёрный треугольничек, и выбрать из выпадающего списка.



Переместить спрайт в точку экрана с заданными координатами X и Y. В точке с заданными координатами окажется центр спрайта. Применение этого блока равносильно применению одновременно двух блоков «Установить X в» и «Установить Y в».



Подробнее о координатах можно узнать в приложении 4.



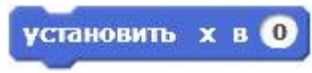
Переместить спрайт в указатель мышки, либо в центр указанного, с помощью выпадающего списка, спрайта. При этом центр спрайта окажется на самом кончике мышки.



Плавно переместить спрайт в точку с заданными координатами. Время перемещения задаётся в секундах. Для быстрого перемещения можно использовать десятичные дроби. Например, 0.5 секунд — это половина секунды, 0.25 секунд — это четверть секунды, 0.1 секунд — это одна десятая секунды, это очень быстро.



Изменить координату X спрайта на указанную величину. При этом спрайт сместиться вправо. При изменении на отрицательную величину, спрайт сместиться влево.



Установить новое значение координаты X спрайта.



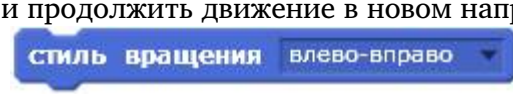
Изменить координату Y спрайта на указанную величину. При этом спрайт сместиться вверх. При изменении на отрицательную величину, спрайт сместиться вниз.



Установить новое значение координаты X спрайта.



Очень важный блок! Позволяет спрайту отразиться от границы окна и продолжить движение в новом направлении.



Изменяет стиль вращения спрайта. Всего есть три типа вращения: вправо-влево, не вращать, кругом. Подробнее о стилях вращения можно узнать в приложении 1. (вставить ссылку)



Это не управляющий блок, а блок получения информации из программы. Он возвращает из памяти работающей программы текущее значение координаты X спрайта. Этот блок, как и все овальные блоки, можно использовать в операторах наравне с числами и переменными.



Блок получения информации из программы. Он возвращает из памяти работающей программы текущее значение координаты Y спрайта.



Блок получения информации из программы. Он возвращает из памяти работающей программы текущее направление спрайта в градусах.

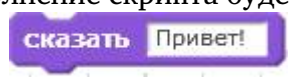


Совет. При двойном клике на любом из блоков, текущий спрайт выполнит указанную команду.

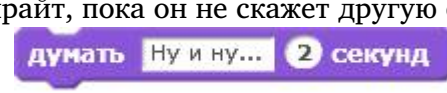
Б. ФИОЛЕТОВЫЕ БЛОКИ ВНЕШНОСТИ.



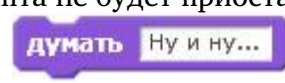
Спрайт будет говорить слово «Привет!» в течение двух секунд. При этом выполнение скрипта будет приостановлено.



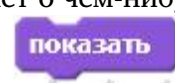
Спрайт постоянно будет говорить слово «Привет!» без остановки выполнения скрипта. Сказанная фраза будет постоянно сопровождать спрайт, пока он не скажет другую фразу.



Спрайт будет думать «Ну и ну…» в течение двух секунд. При этом выполнение скрипта не будет приостановлено.



Спрайт постоянно будет думать «Ну и ну…» без остановки выполнения скрипта. Эта мысль будет постоянно сопровождать спрайт, пока он не подумает о чем-нибудь другом.



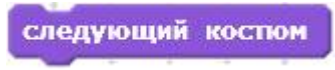
Спрайт станет виден. Если, при выполнении этого блока, спрайт не виден, то, возможно, он скрыт другим спрайтом. В этом случае может помочь блок «Перейти в верхний слой». Также следует проверить координаты спрайта, может быть, он находится за границей экрана.



Спрайт станет невидимым.



Спрайт изменит свой текущий костюм на костюм, выбранный из выпадающего списка.



Спрайт изменит свой текущий костюм на следующий костюм.



Текущий фон сцены изменится на фон, выбранный из выпадающего списка.



Значение эффекта «цвет» изменится на указанную величину. В этом примере значение эффекта увеличится на 25.



Значение эффекта «цвет» изменится до указанного значения. В этом примере значение эффекта станет равно нулю.



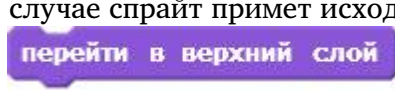
Спрайт примет исходный вид, все графические эффекты будут убраны.



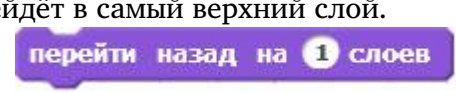
Изменить размер спрайта на указанное количество процентов. В данном случае на 10%.



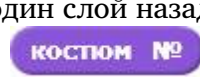
Установить размер спрайта в процентах от исходного размера. В данном случае спрайт примет исходный размер.



Спрайт перейдёт в самый верхний слой.



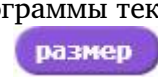
Спрайт перейдёт на один слой назад.



Это не управляющий блок, а блок получения информации из программы. Он возвращает из памяти работающей программы цифровое значение текущего номера костюма.



Блок получения информации из программы. Он возвращает из памяти работающей программы текущее имя фона.



Блок получения информации из программы. Он возвращает из памяти работающей программы текущий размер спрайта.

У сцены есть ещё несколько дополнительных блоков, которых

нет у спрайтов.



Сцена сменит фон на выбранный из выпадающего списка.



Фон сцены изменится на следующий.



Блок получения информации из программы. Он возвращает из памяти работающей программы цифровой номер текущего фона.

В. МАЛИНОВЫЕ БЛОКИ ЗВУКОВ.



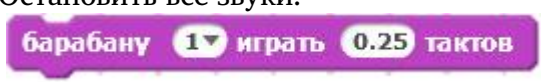
Играть звук «pop», выбранный из библиотеки звуков, без остановки выполнения скрипта.



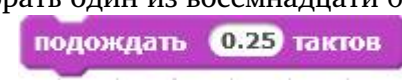
Играть звук «pop», выбранный из библиотеки звуков до завершения. При этом выполнение скрипта будет приостановлено.



Тишина! Остановить все звуки.



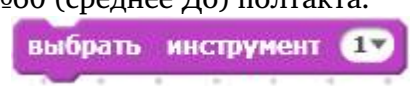
Первому барабану играть одну четверть такта. В выпадающем списке можно выбрать один из восемнадцати барабанов.



Подождать одну четверть такта.



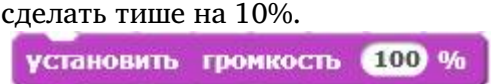
Играть ноту №60 (среднее До) полтакта.



Выбрать один из двадцати музыкальных инструментов с помощью выпадающего списка.



Изменить громкость на указанное количество процентов. В данном случае сделать тише на 10%.



Установить требуемый уровень громкости в процентах от

системного уровня.



Блок получения информации из программы. Он возвращает из памяти работающей программы текущий уровень громкости в процентах.



Изменить темп на 20 ударов в минуту.



Установить темп 60 ударов в минуту.

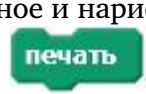


Блок получения информации из программы. Он возвращает из памяти работающей программы текущее значение темпа в ударах в минуту.

Г. ЗЕЛЁНЫЕ БЛОКИ РИСОВАНИЯ ПЕРОМ.



Очищает всё отпечатанное и нарисованное перьями на сцене.



Отпечатывает изображение спрайта на сцене. Не на фоне, а на сцене. Фоны можно изменять, а отпечаток будет поверх всех фонов.



Опустить перо. После этой команды все перемещения пера будут сопровождаться линией.



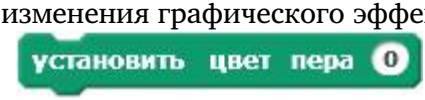
Поднять перо. После этой команды перемещения пера не будут оставлять следов на сцене.



Установить цвет пера. Выбор цвета можно осуществить при помощи курсора.



Изменить цвет пера на 10. Принцип изменения цвета такой же, как и в блоке изменения графического эффекта «цвет».



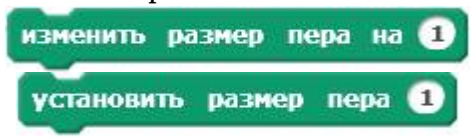
Установить цвет пера в соответствии с таблицей цветов.



Изменить оттенок пера на 10%.

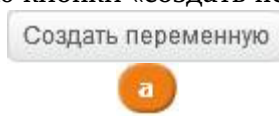


Установить оттенок пера в 50%.



Д. ОРАНЖЕВЫЕ БЛОКИ ДАННЫХ.

Блоки переменных не видны до тех пор, пока вы не создадите переменную с помощью кнопки «создать переменную».



Это сама переменная. При использовании в программе этот миниблок возвращает из памяти работающей программы текущее значение переменной «а». Если около переменной поставить маленькую галочку,



то её значение будет показано на экране.



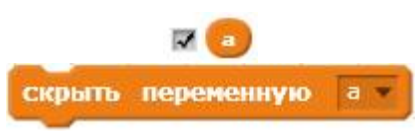
Установить значение переменной «а» в определённое значение. Значение может быть, как числовым, так и текстовым. С помощью выпадающего списка можно выбрать любую переменную.



Изменить значение переменной «а» на единицу. С помощью выпадающего списка можно выбрать любую переменную. Изменять можно только числовую переменную, при попытке изменить текстовую переменную будет ошибка «NaN».



Переменная будет показана на экране, и около неё появится галочка.



Переменная скроется из вида, а маленькая галочка будет

снята.



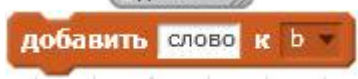
Блоки списков не видны до тех пор, пока вы не создадите список с помощью кнопки «создать список».



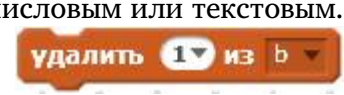


Это сам список. Если около него поставить маленькую галочку, то на экране появится список.





Добавить ещё один элемент списка к списку «b». Элемент списка может быть числовым или текстовым.



Удалить элемент под номером 1 из списка «b».



Вставить значение в элемент списка «b» под номером 1, при этом все последующие элементы сдвинуться. Вставляемое значение может быть либо числовым, либо текстовым.



Заменить значение в элементе списка «b» под номером 1, при этом все последующие элементы останутся на своих местах. Вставляемое значение может быть либо числовым, либо текстовым.



Блок возвращает из памяти работающей программы текущее значение элемента списка «b» под номером 1.



Блок возвращает из памяти работающей программы длину списка «b», то есть количество элементов списка «b».



Логическое значение. Принимает значение «истина», если

список «b» содержит элемент со значением «слово».



Список будет показан на экране, и около него появится галочка.



Список скроется из вида, а маленькая галочка будет снята.

Е. БЛОКИ СОБЫТИЙ.



Блок начала выполнения программы. Выполняется, если нажат зелёный флажок.



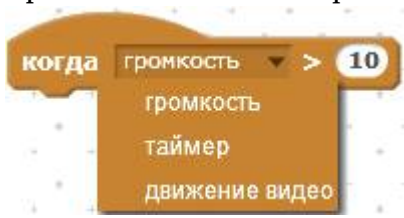
Блок управления с клавиатуры. Выполняется, если нажата клавиша, выбранная из выпадающего списка.



Блок управления мышкой. Выполняется, если мышкой кликнуть на спрайт.



Блок управления скриптом в зависимости от фона. Выполняется, если произошло изменение фона.



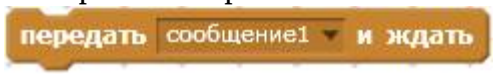
Блок управления скриптом в зависимости от уровня громкости, либо от значения таймера, либо от движения видео. Выполняется при соответствующем изменении параметра, выбранного из выпадающего списка.



Блок управления скриптом с помощью сообщений. Выполняется при получении соответствующего сообщения, выбранного из выпадающего списка.



Блок передачи сообщения, на которое реагирует блок «когда я получу сообщение». При передаче сообщения с помощью этого блока выполнение скрипта не приостанавливается.



Блок передачи сообщения, на которое реагирует блок «когда я получу сообщение». При передаче сообщения с помощью этого блока выполнение скрипта приостанавливается, и он ждёт, пока другие спрайты выполнят все скрипты блока «когда я получу сообщение» до конца. Затем выполнение текущего скрипта продолжается. Чтобы лучше понять, как работает этот блок, потренируемся на кошках. Сделайте Коту следующие два скрипта.



При нажатии на пробел, первый скрипт передаст сообщение «Мяу», и будет ждать, пока не закончится выполнение второго скрипта. Второй скрипт выполняется ровно две секунды. После того, как Кот наорётся, управление снова будет передано первому скрипту, и он скроет Кота с глаз долой, чтобы не шумел по ночам.

Ё. БЛОКИ УПРАВЛЕНИЯ.



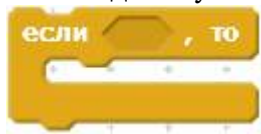
Выполнение скрипта будет приостановлено на указанное количество секунд.



Блок циклического повтора. Указанное количество раз выполняет блоки, которые находятся у него внутри.



Блок постоянного циклического выполнения. Постоянно выполняет блоки, которые находятся у него внутри.



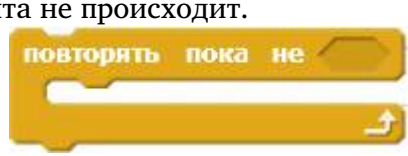
Блок условия «если». Если условие в заголовке блока истинно, то будут выполнены блоки, которые находятся у него внутри.



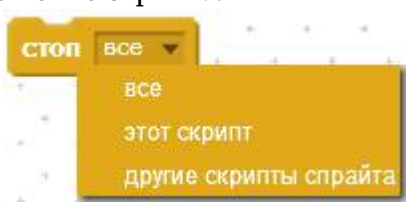
Блок условия «если-иначе». Если условие в заголовке блока истинно, то будут выполнены блоки, которые находятся после слова «если», в противном случае, будут выполнены блоки, которые находятся после слова «иначе».



Блок ожидания выполнения условия. Выполнение скрипта приостанавливается до тех пор, пока условие не станет истинным. Если условие сразу было истинным, то задержки выполнения скрипта не происходит.



Блок цикла с условием. Содержимое блока будет выполняться до тех пор, пока условие в заголовке не станет истинным. То есть цикл работает до тех пор, пока условие в заголовке ложно, как только оно станет истинным, выполнение цикла завершится, и продолжится выполнение скрипта.



Скрипт остановки выполнения. Из выпадающего списка можно выбрать, что остановить. Все скрипты программы, текущий скрипт, в котором использован этот блок или другие скрипты этого спрайта.



Блок управления клонами спрайта. Этот блок выполняется при создании клона спрайта.



Блок создания клонов. Из выпадающего списка можно выбрать, клон какого спрайта необходимо создать, если выбрать «себя самого», то будет создан клон текущего спрайта.



Блок удаляет клон спрайта. Удаление клонов необходимо для того, чтобы их не становилось слишком много. При большом количестве клонов программа может зависнуть.

Ж. БЛОКИ СЕНСОРОВ.



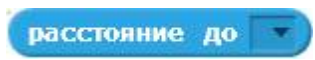
Сенсор касания спрайтов. Из выпадающего списка можно выбрать любой спрайт, а также курсор мышки и край экрана. Сенсор принимает значение «истина», если текущий спрайт касается выбранного.



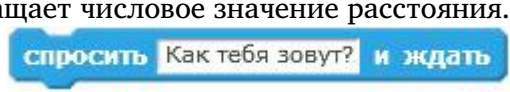
Сенсор касания цвета. Цвет можно выбрать с помощью курсора мышки, который становится похож на руку. Сенсор принимает значение «истина», если текущий спрайт касается выбранного цвета.



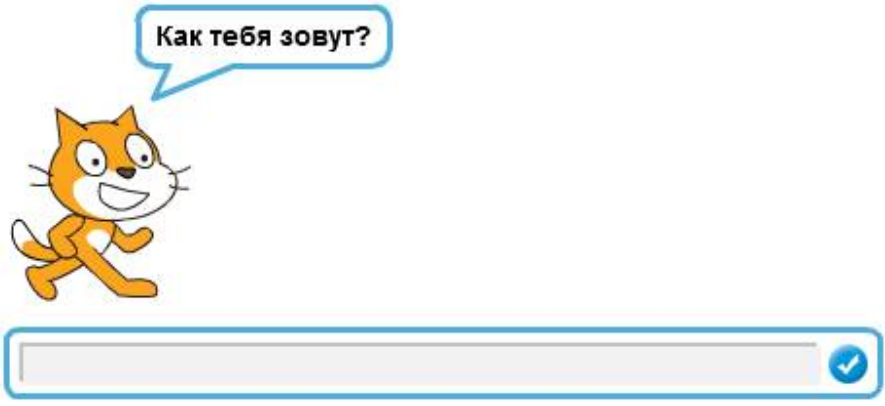
Сенсор касания одного цвета другим. Цвета можно выбрать с помощью курсора мышки, который становится похож на руку. Сенсор принимает значение «истина», если два выбранных цвета соприкасаются.



Блок вычисления расстояния от текущего спрайта, до спрайта, выбранного из выпадающего списка или до указателя мышки. Возвращает числовое значение расстояния.



Блок запроса на ввод информации. Выводит на экран вопрос, и строку ввода информации. После ввода ответа нажмите Enter, или кликните мышкой на синий кружочек с галочкой.



ответ

Блок ввода информации. Возвращает значение, введённое пользователем в строке ввода.

клавиша пробел нажата?

Сенсор нажатия клавиш клавиатуры. Сенсор принимает значение «истина», если нажата клавиша, выбранная из выпадающего списка.

мышка нажата?

Сенсор нажатия мышки. Сенсор принимает значение «истина», если мышка нажата.

мышка по x

Блок возвращает из памяти работающей программы текущую координату X курсора мышки.

мышка по y

Блок возвращает из памяти работающей программы текущую координату Y курсора мышки.

громкость

Блок возвращает из памяти работающей программы текущее значение уровня громкости.

видео движение на этот спрайт

Блок возвращает из памяти работающей программы количество движений над видеоизображением.

повернуть видео на включить

Блок поворота видеоизображения.

установить прозрачность видео 50 %

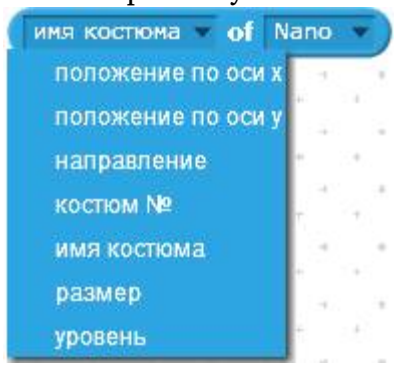
Блок изменения прозрачности видеоизображения.



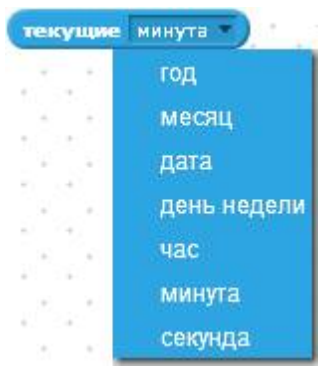
Блок возвращает из памяти работающей программы текущее значение таймера в секундах, то есть время с момента последнего перезапуска таймера.



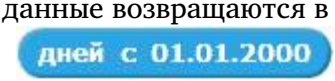
Блок перезапуска таймера. Обнуляет значение таймера.



Блок возвращения из памяти работающей программы различных параметров спрайта, выбранного из выпадающего списка.



Блок возвращения из памяти компьютера текущих значений даты и времени. Все данные возвращаются в числовом формате.



Блок возвращения из памяти компьютера количества дней, прошедших с начала века.



Блок возвращения из памяти сайта scratch.mit.edu имени пользователя, который запустил программу.

3. БЛОКИ ОПЕРАТОРОВ.



Оператор сложения двух чисел.



Оператор вычитания двух чисел



Оператор умножения двух чисел.



Оператор деления двух чисел.



Оператор генерации случайных чисел. Возвращает случайное целое число из заданного диапазона.



Логическое сравнение двух чисел. Возвращает значение «истина», если первое число меньше второго.



Логическое сравнение двух чисел. Возвращает значение «истина», если числа равны.



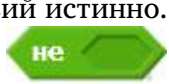
Логическое сравнение двух чисел. Возвращает значение «истина», если первое число больше второго.



Логическая операция И. Возвращает значение «истина», если оба условия истинны.



Логическая операция ИЛИ. Возвращает значение «истина», если хотя бы одно из условий истинно.



Логическое НЕ. Возвращает значение «истина», если условие ложно.



Блок объединения двух текстовых строк. Возвращает объединённую строку.



Блок получения символа из строки текста. Возвращает символ с указанным номером из строки текста.



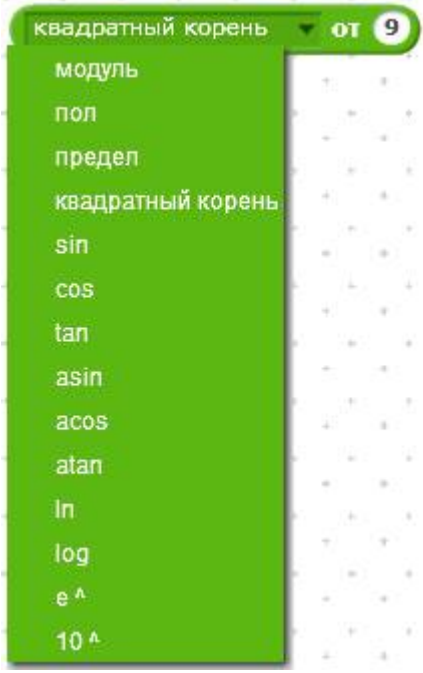
Блок определения длины строки. Возвращает числовое значение длины текстовой строки.



Оператор остаток от деления. Возвращает остаток от деления первого числа на второе.



Оператор округления. Возвращает число без дробной части.

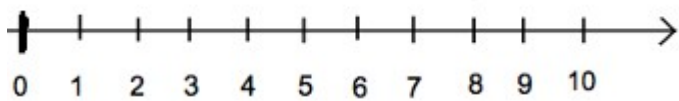


Оператор математических функций. Возвращает числовое значение математической функции, выбранной из выпадающего списка.

ПРИЛОЖЕНИЕ 3. ОТРИЦАТЕЛЬНЫЕ ЧИСЛА.

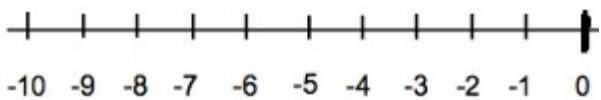
Все Вы умеете считать и знаете, что числа идут по порядку. Каждое следующее число больше предыдущего. Поэтому можно написать все числа в ряд и нарисовать рядом прямую линию. Это числовая ось.

Числовая ось, это прямая линия, на которой написаны все числа, начиная от 0 и до края бумаги. На самом деле числовая ось бесконечная. Если мысленно продлить её за пределы листа бумаги, то на ней будут числа 100 (сто), 1000 (тысяча), 1000000 (миллион), 1000000000 (миллиард) и даже ещё больше.



Числовая ось очень похожа на линейку. Если нарисовать числа через один сантиметр, то можно что-нибудь измерить.

Все числа на такой числовой оси больше ноля. Такие числа называются «положительные». Они вызывают только положительные эмоции. Например, два арбуза, три персика, пять шоколадок. А ещё бывают «отрицательные» числа. Это числа со знаком «минус».



Они идут от ноля влево до края бумаги и ещё дальше. Бывают числа -100, -1000, -1000000. Отрицательные числа вызывают отрицательные эмоции.

Например, если я дам Вам минус три персика, то это означает, что я отберу у Вас три персика.

Отрицательные числа обычно пишут в скобках. Вот несколько примеров.

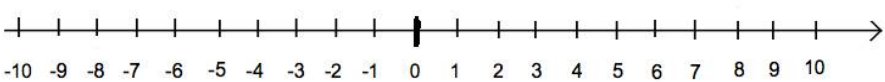
$1+(-1)=0$

$5+(-2)=3$

$(-10)+(-10)=(-20)$

$5+(-10)=(-5)$

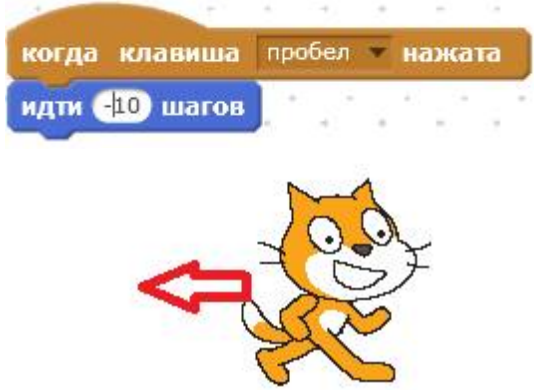
Вот числовая ось целиком.



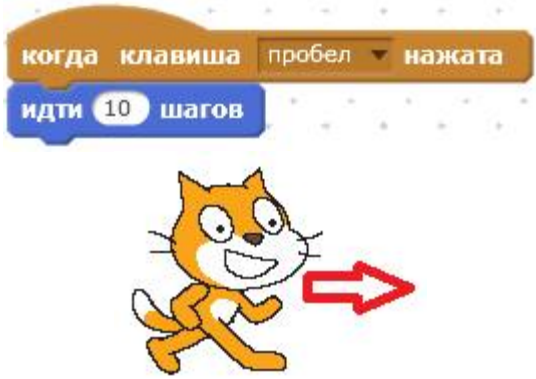
Если мы прибавляем положительное число, мы двигаемся по числовой оси вправо. $2+2=4$. 4 правее, чем 2. Если мы

прибавляем отрицательное число, то мы двигаемся по числовой оси влево. 4 + (-8) = (-4). (-4) левее чем 4.

Отрицательные числа очень часто используются в создании игр. Любое движение влево — это отрицательное число.



Движение вправо — это положительное число.



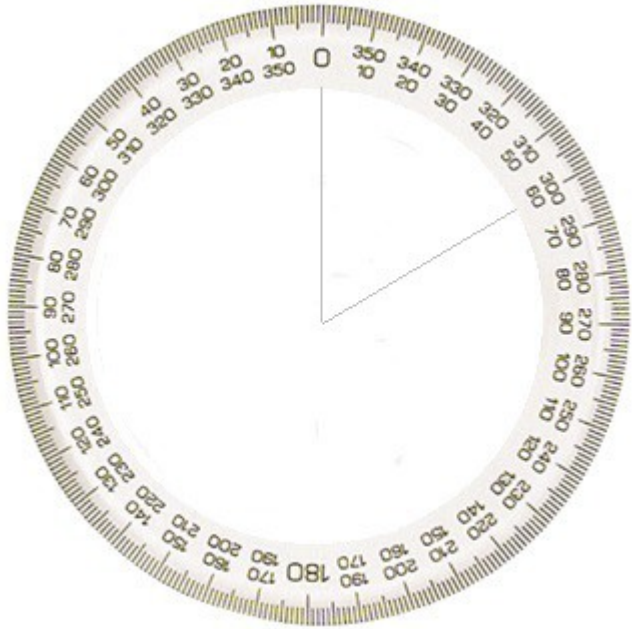
Сделайте такие программки и нажимайте пробел. Увидите куда пойдёт Кот.

ПРИЛОЖЕНИЕ 4. ГРАДУСЫ.

Все Вы знакомы с часами. Циферблат делится на 12 частей. Две стрелки образуют угол с вершиной в центре круга.



Вверху 0 часов, или 12. Справа 3 часа. Внизу 6 часов. Слева 9 часов. В математике подобным образом круг делится на 360 градусов. На следующем рисунке посмотрите на нижний ряд цифр, который идёт по часовой стрелке — это величина угла в градусах. Верхний ряд цифр — отрицательные градусы, которые отсчитываются против часовой стрелки. На рисунке нарисован угол величиной 60 градусов с вершиной в центре круга.



В Scratch, как и в математике, круг делится на 360 градусов. В Scratch мы не рисуем углы, а используем их для определения направления движения спрайтов.

Направление 0 градусов это вверх.

Направление 90 градусов это вправо.

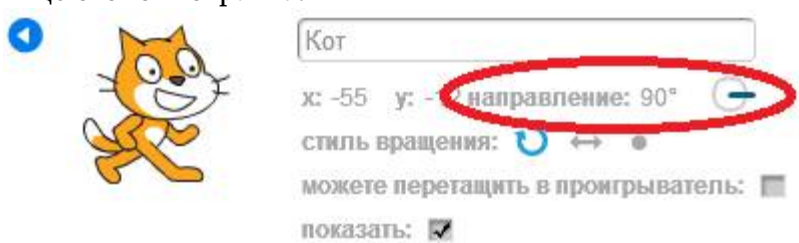
Направление -90 градусов это влево.

Направление 180 градусов это вниз.

Если Вы внимательно посмотрите на рисунок, то увидите, что -90 градусов равно 270 градусов, а 180 градусов равно минус 180 градусов.

Обратите внимание, на оси x значение 180 не равно -180! 180 находится справа, а -180 слева от начала координат. На круге же, это одно и то же, 180 это вниз, и -180 тоже вниз.

Направление движения спрайта всегда можно увидеть на странице свойств спрайта.



Сделаем на память несколько фотографий Кота в разных направлениях.

Кот в направлении 90 градусов.



Кот в направлении 0 градусов.

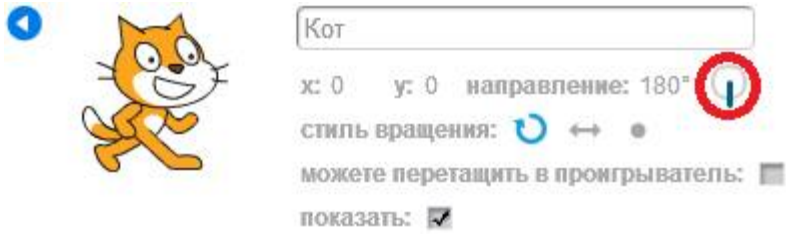


Кот в направлении -90 градусов.



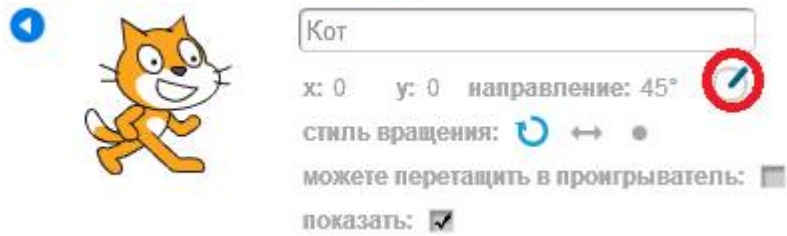
Кот в направлении 180 градусов.





Кот в направлении 45 градусов.





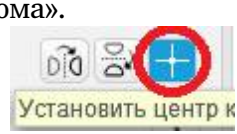
Кот в направлении 135 градусов.



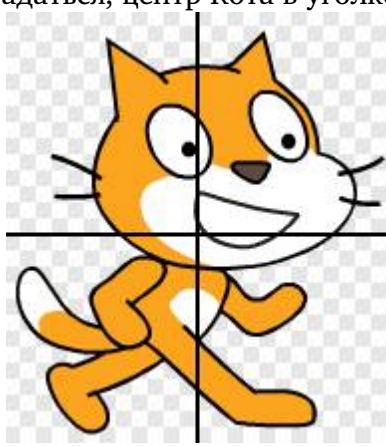
ПРИЛОЖЕНИЕ 5. КООРДИНАТЫ X И Y.

Экран проекта в Scratch состоит из пикселей — маленьких точек. Ширина экрана 480 пикселей, высота экрана 360 пикселей. Для того чтобы точно знать где на экране находится каждый спрайт, все точки на экране пронумерованы. Но как же понять, в какой точке экрана находится спрайт, ведь он не является точкой!

Да, спрайт не является точкой, но у него есть точка привязки, это та точка, координаты которой и являются координатами всего спрайта. Посмотрите где точка привязки у Кота, для этого зайдите в графический редактор и кликните на кнопке «Установить центр костюма».

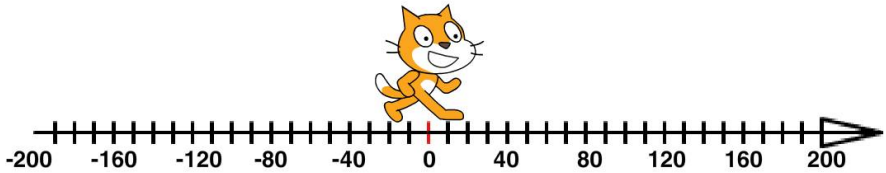


Кто бы мог догадаться, центр Кота в уголке его рта!

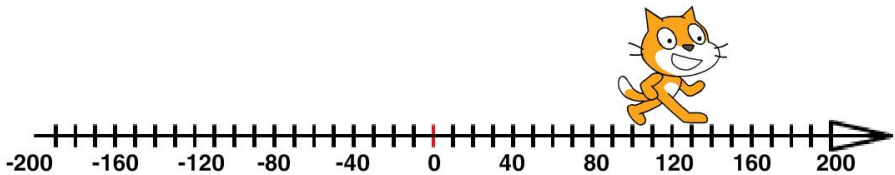


Теперь, каждый раз, когда вы увидите, что Кот находится в точке с некоторыми координатами, знайте, что в этой точке находится уголок его ротика, а сам Кот расположен вокруг этой точки.

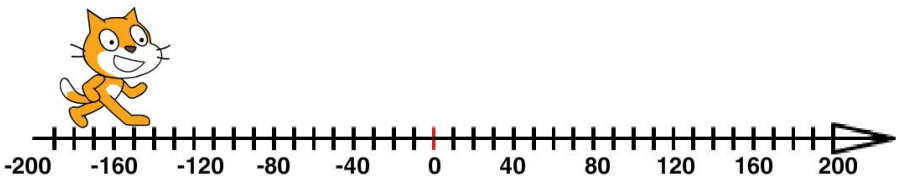
Начало отсчёта координат — центр экрана. Каждой точке экрана соответствуют две координаты X и Y. Координата X показывает, на сколько пикселей левее или правее от центра находится спрайт. На рисунке показана координатная ось X, она проходит точно по центру экрана. Если спрайт находится справа от центра экрана, то его координата X положительная, если спрайт левее центра, то его координата X отрицательная. На рисунке координата X Кота равна нулю.



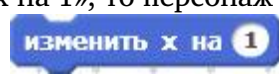
На следующем рисунке координата икс Кота равна 120. X = 120.



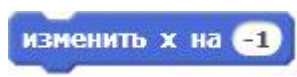
На следующем рисунке координата икс Кота равна -160. X = -160.



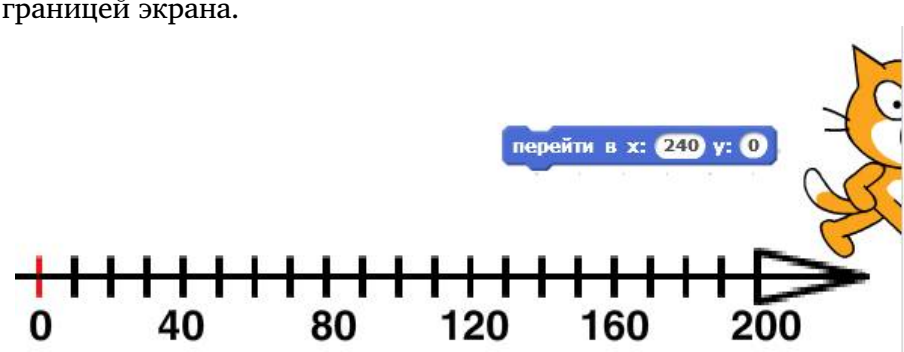
Как видите, на оси X положительные числа начинаются от ноля и увеличиваются вправо, а отрицательные числа начинаются тоже от ноля и увеличиваются влево, значит, если мы пишем в программе «изменить x на 1», то персонаж двигается вправо.



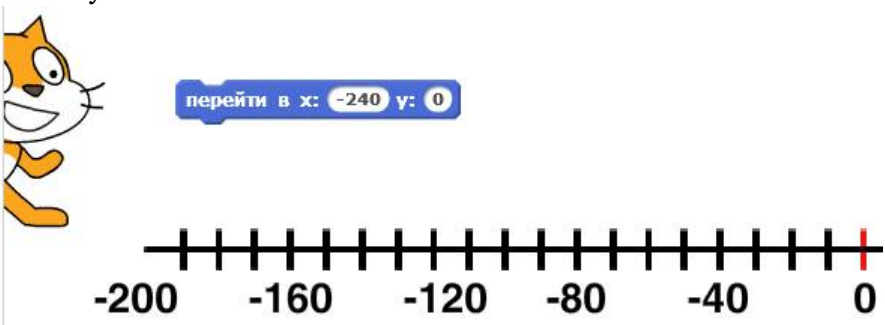
Если мы пишем в программе «изменить x на -1», то персонаж двигается влево.



Таким образом, координата X отвечает за перемещение спрайтов вправо и влево. Крайние точки на экране X = -240 слева и X = 240 справа. Если Кота отправить на край экрана блоком «идти в X = 240 Y = 0», то Кот наполовину спрячется за правой границей экрана.

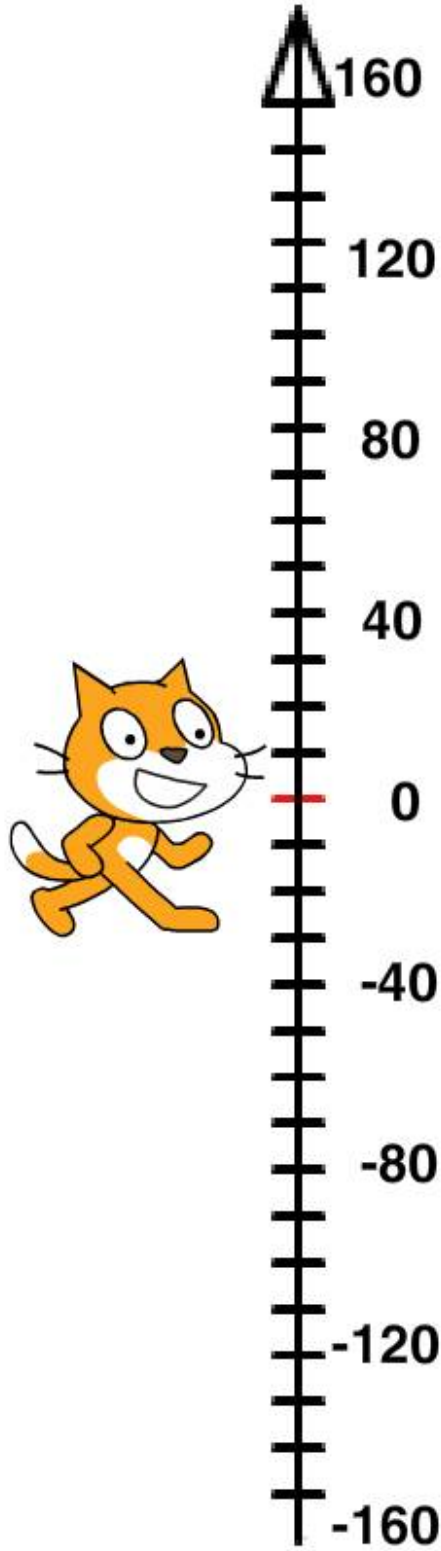


Если Кота отправить в X = -240, то Кот торчит только на половину слева.

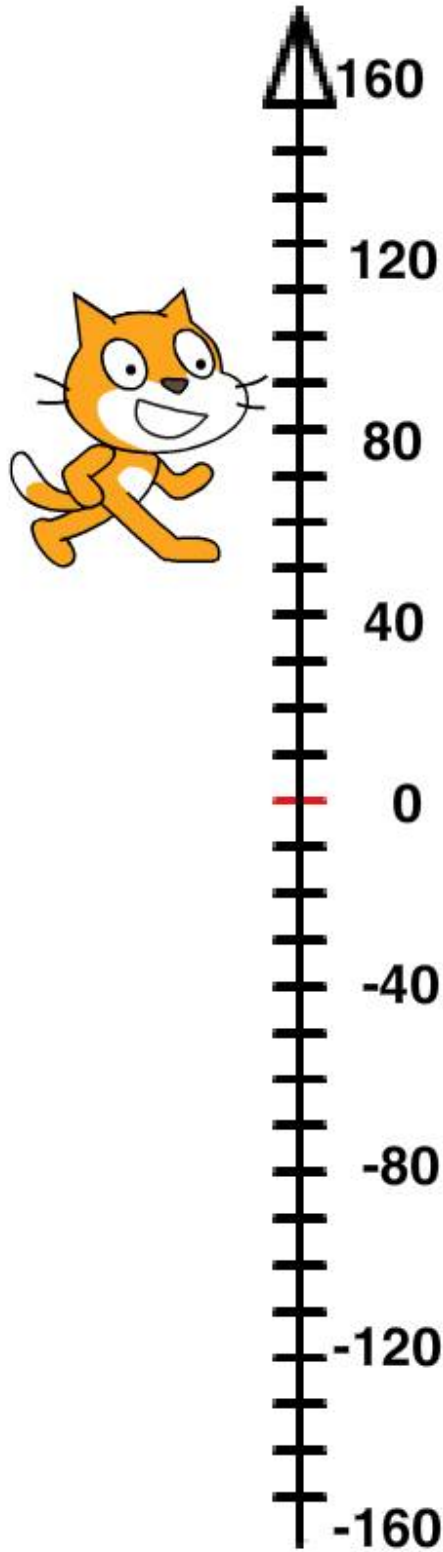


За вертикальные перемещения спрайтов отвечает ось Y (читается ось игрек). На рисунке показана координатная ось Y, она проходит точно по центру экрана. Если спрайт находится выше центра экрана, то его координата Y положительная, если спрайт ниже центра, то его координата Y отрицательная.

На следующем рисунке координата Y Кота равна нулю.


160
120
80
40
0
-40
-80
-120
-160

На следующем рисунке координата игрек Кота равна 80. Y = 80.


160
120
80
40
0
-40
-80
-120
-160

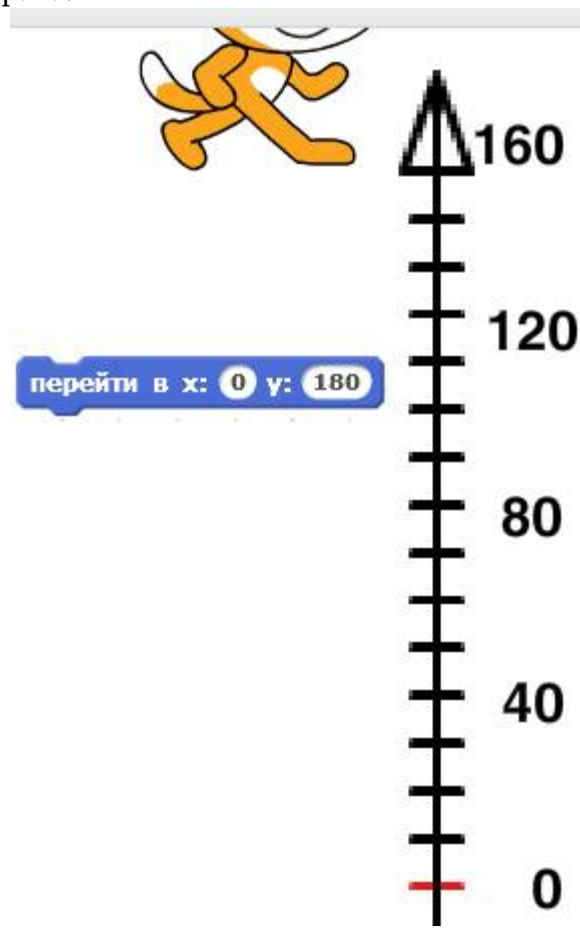
Как видите, на оси Y положительные числа начинаются от ноля и увеличиваются вверх, а отрицательные числа начинаются тоже от ноля и увеличиваются вниз, значит, если мы пишем в программе «изменить Y на 1», то персонаж двигается вверх.



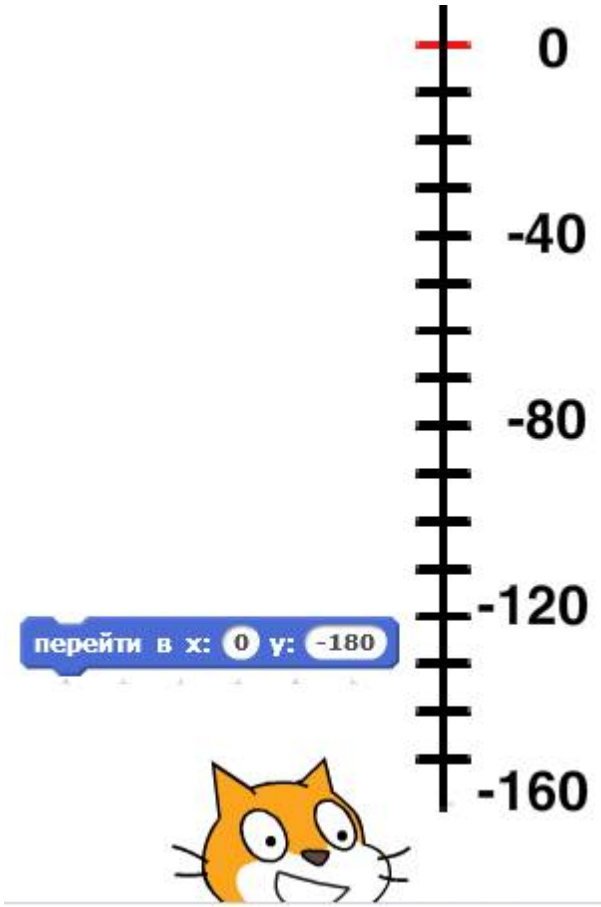
Если мы пишем в программе «изменить Y на -1», то персонаж двигается вниз.



Таким образом, координата Y отвечает за перемещение спрайтов вверх и вниз. Крайние точки на экране Y = -180 внизу и Y = 180 наверху. Если Кота отправить на край экрана блоком «идти в X = 0 Y = 180», то Кот наполовину спрячется за верхней границей экрана.



Если Кота отправить на край экрана блоком «идти в X = 0 Y = -180», то Кот наполовину спрячется за нижней границей экрана. Только усы торчат.



Если менять одновременно и икс, и игрек, то кота можно отправить в любую часть экрана. Например, в верхний правый угол «идти в X = 180 Y = 120».



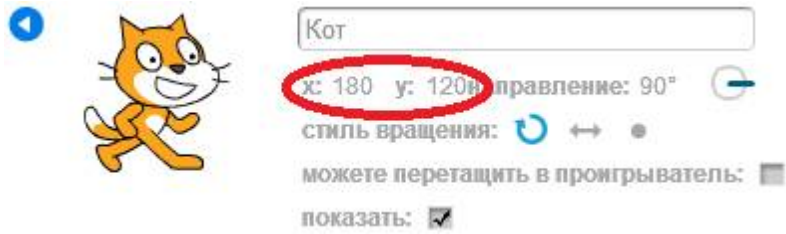


Левый нижний угол «идти в X = -180 Y = -120»



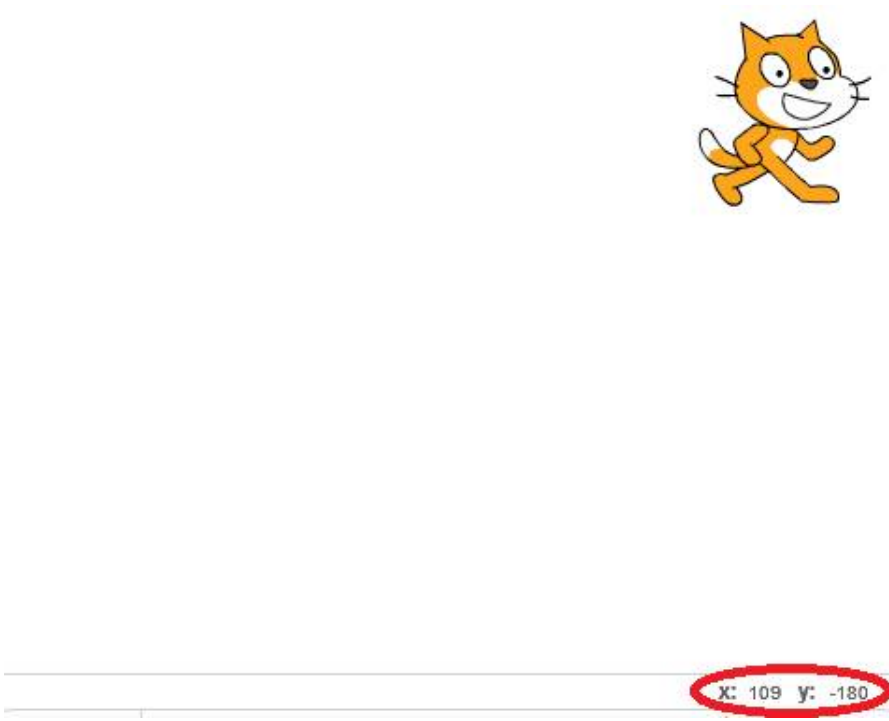


Взглянув на свойства спрайта, мы всегда можем определить, в какой части экрана он находится.



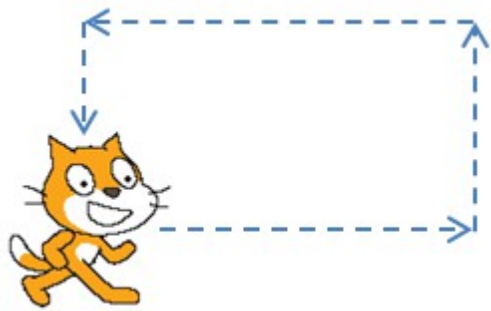
Спрайт в точке с координатами X = 180 и Y = 120, значит кот в правом верхнем углу.

Текущие координаты курсора мышки можно видеть под окном справа. Когда мышка двигается, числа X и Y постоянно изменяются.



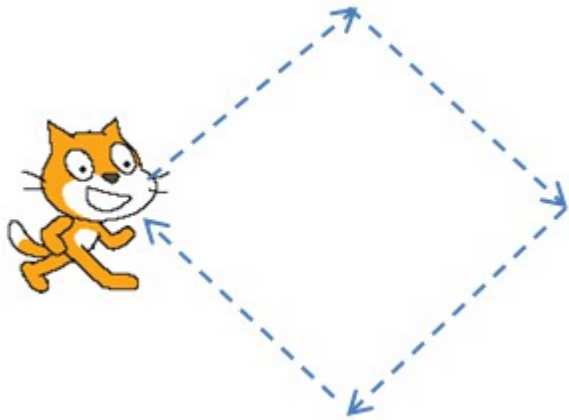
Попробуйте поставить мышку точно в точку X = 0 Y = 0.

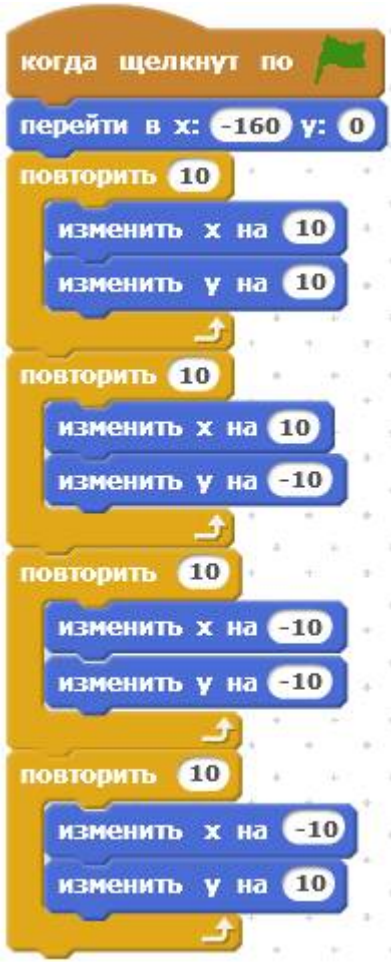
Примеры использования координат. Кот проедет по прямоугольно траектории.





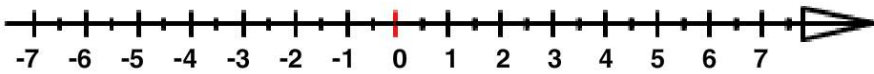
Если одновременно изменять x и y, то Кот будет двигаться по диагонали.



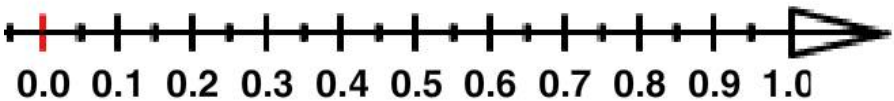

когда щелкнут по
перейти в x: -160 y: 0
повторить 10
изменить x на 10
изменить y на 10
повторить 10
изменить x на 10
изменить y на -10
повторить 10
изменить x на -10
изменить y на -10
повторить 10
изменить x на -10
изменить y на 10

ПРИЛОЖЕНИЕ 6. ДЕСЯТИЧНЫЕ ДРОБИ.

В приложении 2 Вы уже видели числовую ось. Числа идут от ноля и до бесконечности вправо, и от ноля до бесконечности влево (отрицательные числа).



Если числовую ось приблизить и рассмотреть внимательно, то окажется что между целыми числами 0, 1, 2…, есть дробные числа. Например, между нулём и единицей поместилось 9 дробных чисел. Они очень похожи на простые числа, только в 10 раз меньше. Например, 0,1 в 10 раз меньше чем 1.



Это можно использовать, чтобы замедлить движение персонажа.

Сделайте Коту такую программу.



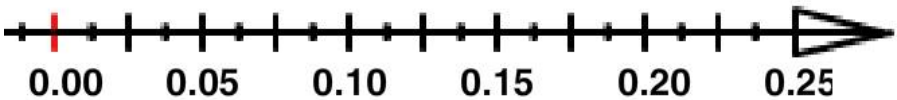
Нажмите пробел. Кот двигается медленно. Как заставить двигаться его ещё медленнее? Измените 1 на 0.5. Двигается в 2 раза медленнее.



Измените 0.5 на 0.1. Двигается ещё в 5 раз медленнее.



Если числовую ось еще приблизить, то появятся еще более мелкие числа.



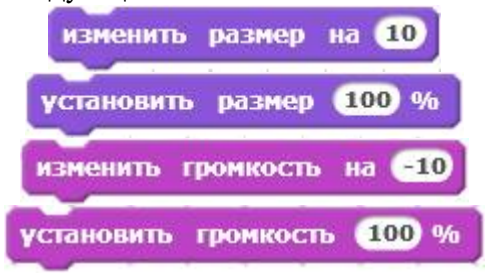
Можно записать математически $1/2 = 0.5$ $1/4 = 0.25$ $1/10 = 0.1$

0.5 в два раза меньше единицы, 0,25 в четыре раза меньше единицы, 0.1 в десять раз меньше единицы.

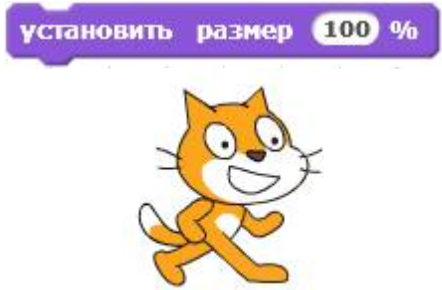
Мы будем очень часто использовать дробные числа в программах.

ПРИЛОЖЕНИЕ 7. ПРОЦЕНТЫ.

Процентом называется одна сотая часть от чего-либо. Процент обозначается символом %. Проценты в Scratch используются в следующих блоках.



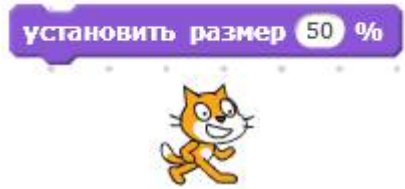
Например, блок «установить размер 100%» означает установить нормальный размер спрайта.



Блок «установить размер 200%» означает сделать спрайт в два раза больше.



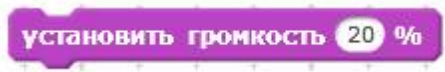
Блок «установить размер 50%» означает сделать спрайт в два раза меньше.



Блок «установить громкость 100%» означает установить нормальную громкость.

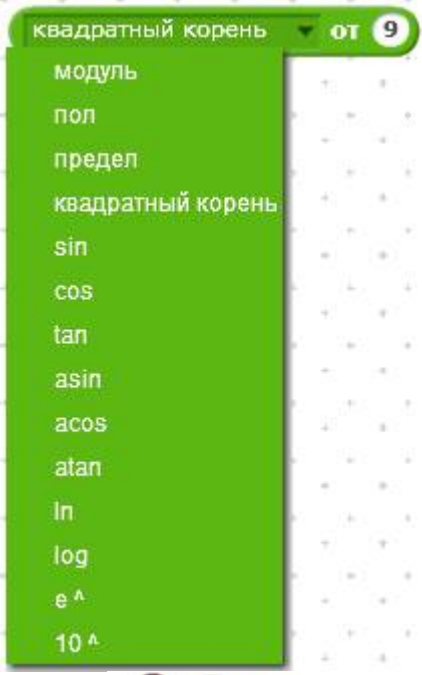


Блок «установить громкость 20%» означает установить громкость в пять раз тише нормальной.



ПРИЛОЖЕНИЕ 8. ФУНКЦИИ.

Все математические функции Scratch собраны в один блок с помощью выпадающего списка. Рассмотрим их по порядку.



Совет. Двойной клик на функции покажет её результат, например.



МОДУЛЬ.

Модуль — это абсолютное значение числа. Модуль 5 равен 5. Модуль -5 тоже равен 5. То есть модуль отбрасывает от всех чисел минусы, и делает все числа положительными. В математике модуль записывается с помощью вертикальных палочек, в других языках программирования модуль называется abc().

Например

$|5| = 5$



$|-5| = 5$



ОКРУГЛЕНИЕ ЧИСЕЛ.

Для округления числа используют две функции «floor» и «ceiling». В Scratch названия этих функций перевели с ошибкой, и получилось «пол» и «предел». На самом деле надо было перевести так «floor» - «округление в меньшую сторону», а «ceiling» - «округление в большую сторону». Эти функции нужны чтобы из дробных чисел делать целые числа.

Функция «пол» округляет в меньшую сторону, просто отбрасывает все цифры, после точки.

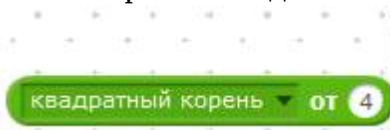


Функция «предел» округляет в большую сторону, увеличивает число до следующего целого.

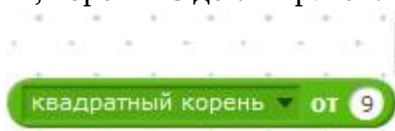


КВАДРАТНЫЙ КОРЕНЬ.

Эта функция определяет, какое число нужно умножить на само себя, чтобы получился результат. Например, если два в квадрате равно четырём, то есть два умножить на два равно четыре, то корень из четырёх — это два.



$3*3 = 9$, значит, корень из девяти равен?



ТРИГОНОМЕТРИЧЕСКИЕ И ЛОГАРИФМИЧЕСКИЕ ФУНКЦИИ.

Остальные функции нам пока не нужны. Их изучают в старших классах. Они используются для программирования физики в играх.

КРАТКОЕ СОДЕРЖАНИЕ СЛЕДУЮЩЕЙ КНИГИ.

В следующей книге приведены примеры ещё более сложных и интересных проектов на Scratch.

А также, впервые в этой части галактики!

Каждая игра сопровождается комиксом!!!

ИГРА «ДИВЕРСАНТ».





ИГРА «ФЕРМА».



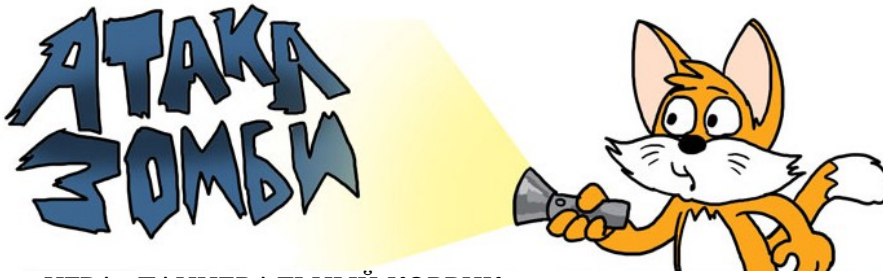


ИГРА «БИТВА ЗА ДЕНЬ РОЖДЕНИЯ КОТА».





ИГРА «АТАКА ЗОМБИ».



ИГРА «ТАНЦЕВАЛЬНЫЙ КОВРИК».



ИГРА «КОСМИЧЕСКАЯ БИТВА».



ИГРА «СОБИРАЕМ ЯБЛОЧКИ».



ИГРА «СТРИТРЕЙСИНГ».

